

Títol: Implementació del procés d'utilització de patrons de requisits

Volum:1 de 1

Alumne: Cristina Palomares Bonache

Director/Ponent: Carme Quer Bosor

Departament: LSI

Data: 25/01/2010

DADES DEL PROJECTE

Títol del Projecte: Implementació del procés d'utilització de patrons de requisits

Nom de l'estudiant: Cristina Palomares Bonache

Titulació: Enginyeria Informàtica

Crèdits: 37,5

Director/Ponent: Carme Quer Bosor

Departament: LSI

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President: Franch Gutiérrez, Xavier

Vocal: Perelló Muntan, Jordi

Secretari: Quer Bosor, Carme

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Implementació del procés d' utilització de patrons de requisits

Agraïments

Aquest espai està dedicat no només a aquells que m’han ajudat a acabar el PFC sinó també a la gent que ha estat amb mi durant aquests últims 5 anys de la meva vida com a estudiant a la universitat i m’han ajudat a arribar aquí i, d’alguna manera, a ser com sóc ara. Com s’acostuma a dir “Això, va per tots vosaltres!”.

Pels amics de la universitat: a la Jessica Martínez, per la quantitat de pràctiques i treballs que hem fet juntes durant la carrera i per l’inoblidable projecte de PROP ;-), a la Verónica López per les sessions de biblioteca i micros que hem patit els últims dos anys, especialment en època d’exàmens, i per donar-me la possibilitat de guanyar una amiga. A la Sara Dueñas, l’Albert González, el Francesc Navarro i l’Àlex Perea, perquè sense ells els últims projectes de la carrera i els matins de cafè no haurien sigut el mateix.

Pels meus companys de despatx, més conegut com OmegaCIX: l’Oriol, el Borja, la Leo, el David, en Marc i l’Hugo, per les bones estones que hem passat en l’últim any i que t’ajuden a desconnectar de la feina.

Pels desenvolupadors de la primera versió de RPToolManager, l’Oriol Arcas i l’Oriol Collell, especialment a aquest últim, per respondre a tots els meus dubtes sobre l’aplicació i ajudar-me a configurar el sistema per poder començar a desenvolupar.

Pels caps, la Carme Quer i el Xavi Franch, per respondre tots els meus dubtes i preguntes, i per oferir-me l’opció de fer aquest projecte de recerca, treballar amb ells i formar part del grup de recerca GESSI.

I bé, tot i que apareixen en últim lloc i no per això són menys importants, per la meva família i per la meva parella, per ajudar-me en aquests cinc anys a aguantar la tensió i els nervis, per estar sempre i passi el que passi al meu costat, i perquè els petits detalls i les bones estones, encara que semblin ximpleries, són les que realment s’aprecien.

A tots, de tot cor, moltes moltes gràcies!

Índex

1. Introducció.....	11
1.1. Introducció	12
1.2. Motivacions	12
1.3. Organització de la memòria.....	13
2. Patrons de requisits.....	15
2.1. Introducció	16
2.2. Catàleg de Patrons de Requisits.....	16
2.3. Estructura d'un Patró de Requisit.....	17
2.4. Mètode PABRE: Creació i utilització de patrons.....	20
2.5. RPToolManager i RPProjectManager dins del mètode PABRE.....	26
3. Eines de Treball.....	29
3.1. Tecnologies utilitzades	30
3.2. Entorn de Desenvolupament (IDE)	32
3.3. SQuirreL-SQL Client: Vista i modificació d'una Base de Dades existent.....	33
4. Arquitectura: En tres capes	35
4.1. Introducció	36
4.2. Capes i nivells.....	36
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software.....	41
5.1. Procés de manteniment i extensió de software.....	42
5.2. Tipus de manteniment de software	44
6. Anàlisi de la 1ª versió de RPToolManager	47
6.1. Propòsit de l'anàlisi	48
6.2. Desenvolupament de l'anàlisi	48
6.3. Conclusions	49
7. Model Conceptual	61
7.1. Model conceptual de la 1ª versió.....	62
7.2. Nou model conceptual	66
8. Base de Dades.....	67
8.1. Introducció	68
8.2. Modificació del mapeig per Hibernate.....	68
8.3. Modificació de la Base de Dades existent.....	73
9. Segona versió de RPToolManager	75
9.1. Modificació del codi font de l'aplicació.....	76
9.2. Proves del Software.....	76
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager.	79
10. Metodologia: Àgil	81
10.1. Introducció	82
10.2. Breu història.....	82
10.3. Els principis de la metodologia àgil.....	83
10.4. Per què les metodologies àgils funcionen?	83
11. Anàlisi de Requeriments	85
11.1. Propòsit de l'anàlisi de requeriments	86
11.2. Abast del projecte	86
11.3. Requeriments funcionals	87

11.4. <i>Requeriments no funcionals</i>	92
12. Model de Casos d'ús.....	97
12.1. <i>Introducció</i>	98
12.2. <i>Diagrames de Casos d'ús</i>	100
12.3. <i>Descripció dels Casos d'ús</i>	103
13. Model Conceptual	125
13.1. <i>Diagrama de classes</i>	126
13.2. <i>Restriccions d'integritat textuais</i>	129
13.3. <i>Matriu de traçabilitat</i>	130
14. Model de Comportament	133
14.1. <i>Introducció</i>	134
14.2. <i>Visualització de projectes i patrons</i>	135
14.3. <i>Construcció de projectes</i>	141
14.4. <i>Gestió de projectes</i>	162
14.5. <i>Interoperabilitat</i>	164
15. Disseny de la Interfície	167
15.1. <i>Introducció</i>	168
15.2. <i>Disseny de les finestres</i>	168
15.3. <i>Navigation Maps</i>	184
16. Reutilització de Components	189
16.1. <i>Introducció</i>	190
16.2. <i>Reutilització dels components per la finestra principal</i>	190
16.3. <i>Reutilització dels components per a la visualització dels arbres</i>	192
16.4. <i>Reutilització dels components per la finestra d'edició de projectes</i>	193
17. Base de Dades.....	195
17.1. <i>Mapeig per Hibernate</i>	196
17.2. <i>Consideracions sobre el mapeig</i>	198
18. Document de Disseny.....	201
18.1. <i>Visualització de projectes i patrons</i>	202
18.2. <i>Construcció de projectes</i>	212
18.3. <i>Gestió de projectes</i>	232
18.4. <i>Interoperabilitat</i>	234
19. Proves del Software.....	239
19.1. <i>Introducció</i>	240
19.2. <i>Comprovació de les funcionalitats en el curs normal d'esdeveniments</i>	240
19.3. <i>Comprovació de les restriccions d'integritat</i>	242
20. Planificació i Execució del Projecte	245
20.1. <i>Planificació</i>	246
20.2. <i>Execució</i>	248
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
ANNEX.....	263
ANNEX 1: Modificacions al codi font de RPToolManager	265
<i>Errors a nivell de domini</i>	265
<i>Errors a nivell d'interfície</i>	268
<i>Mancances</i>	287
<i>Noves funcionalitats</i>	294

1. Introducció

1. Introducció.....	11
1.1. Introducció	12
1.2. Motivacions	12
1.3. Organització de la memòria.....	13
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1 ^a versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
 20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

1.1. Introducció

Donat a la pressió existent per obtenir software de qualitat i en el menor temps possible que es viu en els últims anys, des de ja fa temps s'han introduït tècniques de reutilització en el procés de desenvolupament de software. Aquestes tècniques faciliten que els components es dissenyin i es desenvolupin amb l'objectiu de poder ser reutilitzats en altres aplicacions, reduïnt el seu temps de desenvolupament, millorant la qualitat del producte i sent més competitius en cost (a menys hores dedicades durant el procés de desenvolupament, evidentment menys cost). Així doncs, el concepte de reutilització en l'etapa d'enginyeria de requisits és un objectiu desitjable des de ja fa anys.

Amb aquest objectiu de reutilització de requisits, d'entre totes les possibles tècniques proposades per aconseguir-la, el grup de recerca GESSI, amb el qual col·laboro des de fa una mica més d'un any, s'ha centrat en l'ús de patrons de requisits. Exactament, s'ha centrat en l'ús d'aquests patrons en l'àmbit de projectes de selecció de components software mitjançant el mètode call-for-tenders.

El motiu de l'ús de patrons és la percepció que tenim de que l'ús d'aquests pot ajudar a resoldre un dels problemes que habitualment existeixen als llibres de requisits: definicions de requisits ambigües, incompletes, incoherents i normalment expressades de forma poc sistemàtica. L'ús de patrons permetrà crear una forma estàndard en l'elaboració i redacció de requisits, i també ajudarà a expressar els requisits de forma general i clara, millorant així la qualitat d'aquests, i reduirà el temps i l'esforç de l'elicitació, portant a estalvis econòmics.

Així doncs, aquest PFC tracta sobre la gestió i l'ús de patrons de requisits i està dividit en dues parts ben diferenciades. La primera s'encarrega de la millora de l'aplicació *RPToolManager* (abreviació de *Requirement Pattern Tool Manager*): una aplicació per a la definició de patrons de requisits i la gestió d'un repositori per aquests de la que el grup GESSI en disposa d'una primera versió. La segona part del PFC es centra en l'ús dels patrons que conté el repositori, i consisteix en el desenvolupament de la primera versió de l'aplicació *RPPProjectManager*: una aplicació que permetrà instanciar els patrons de requisits en projectes de selecció i generarà el document necessari pel call-for-tenders, a més de permetre la gestió de projectes de selecció i navegar pel repositori de patrons.

1.2. Motivacions

El meu primer contacte amb els patrons de requisits va ser fa una mica més d'una any, durant la meua primera col·laboració amb el grup de recerca GESSI; aquesta primera col·laboració va consistir en realitzar una part de la capa de domini i dades de l'aplicació *RPToolManager*, més concretament, en realitzar el mapeig de tot el model conceptual de l'aplicació utilitzant Hibernate.

Fa només uns mesos, cap al juliol del 2009, parlant del que seria el meu futur després d'acabar la carrera amb la que ara és la meva directora de PFC, la Carme Quer, va sorgir la possibilitat de fer un doctorat. D'entre els temes que em van oferir, els Patrons de Requisits va ser el que més m'interessava i m'agradava i vaig pensar que seria una bona idea que el PFC em servís com un primer contacte seriós amb el tema de patrons. Així que què millor primer contacte que familiaritzar-me amb la literatura ja existent i l'eina de la que el grup de recerca GESSI ja disposava. Així que aquí estic, modificant la primera versió de l'aplicació *RPToolManager* i creant la primera de l'aplicació *RPPProjectManager*, el que, com era esperat, ha augmentat el meu coneixement sobre Patrons de Requisits, m'ha permès familiaritzar-me amb la lectura de papers i llibres sobre el tema i ha millorat la meva programació.

1.3. Organització de la memòria

La present memòria està dividida en 24 capítols. El 1r conté la introducció, les motivacions i la present organització de la memòria. En el 2n capítol es dona una explicació sobre els patrons de requisits i el mètode PABRE, i també com encaixen les aplicacions *RPToolManager* i *RPPProjectManager* dins d'aquest últim, per tal que el lector pugui entrar en el context del projecte que s'ha desenvolupat. En el 3r capítol s'explica les eines de treball utilitzades, ja siguin tecnologies usades, llenguatges i APIs de programació implantats o l'entorn de desenvolupament. En el 4t capítol es descriu l'arquitectura en tres capes, arquitectura utilitzada en les dues aplicacions que formen part del projecte desenvolupat.

A partir del 5è capítol i fins al 9è (ambdós inclosos) parlarem del desenvolupament de la 2^a versió de l'aplicació *RPToolManager*. El 5è descriu quins són els punts que abasta el manteniment i l'extensió d'un projecte software, les etapes definides i els quatre tipus de manteniment que estan descrits. En el 6è capítol s'explica quin és l'anàlisi realitzat sobre la 1^a versió de *RPToolManager* i quins van ser els resultats obtinguts. En el capítol 7 es descriu el model conceptual, on també s'expliquen els canvis que ha sofert aquest per la nova versió de l'eina. Al 8è capítol s'expliquen les modificacions realitzades a la Base de Dades de l'eina per adaptar-la a la versió nova. El 9è capítol conté l'explicació de la fase d'implementació de la 2^a versió de *RPToolManager*.

Del capítol 10 al 19 parlarem del desenvolupament de la 1^a versió de l'aplicació *RPPProjectManager*. Dins del capítol 10 es descriu la metodologia utilitzada en el projecte, l'àgil, i es dona una breu descripció així com arguments a favor que expliquen perquè aquest tipus de metodologies funcionen correctament. És a partir del capítol 11 quan es comencen a descriure artefactes propis de l'Enginyeria del Software, en concret aquest capítol tracta de l'anàlisi de requeriments, on es descriu l'obtenció i descripció dels requeriments funcionals i no funcionals. Com a etapa posterior a aquesta fase, trobem en el capítol 12 el model de casos d'ús extrets a partir d'aquest. A partir d'aquí, ja es pot descriure en el capítol 13 el model conceptual i al capítol 14 el model de comportament. En el 15 es pot observar el disseny de la interfície de l'aplicació i en el capítol 16 es descriuen els components reutilitzats a l'aplicació. En el capítol 17è es troba el disseny de la Base de Dades. En el següent, el 18è, tenim el document de

disseny, i amb el 19è capítol, les proves del software, s'acaben els artefactes propis del desenvolupament software.

És a partir del 20è capítol on comencen els artefactes de tipus projecte. En el 20è tenim la planificació i seguiment del projecte. El future work es situa al capítol 21. En el 22 trobem les conclusions del projecte i al 23 la bibliografia emprada. Finalment, el capítol 24 conté l'explicació de quin és el contingut del cd que s'adjunta amb aquesta memòria.

2. Patrons de requisits

1. Introducció.....	11
2. Patrons de requisits.....	15
2.1. Introducció	16
2.2. Catàleg de Patrons de Requisits.....	16
2.2.1. Classification schemas.....	16
2.3. Estructura d'un Patró de Requisit.....	17
2.3.1. Meta-dades del patró	18
2.3.2. Requirement forms.....	18
2.3.3. Fixed part	19
2.3.4. Extensions	19
2.3.5. Form text	19
2.3.6. Question text	20
2.3.7. Dependències	20
2.4. Mètode PABRE: Creació i utilització de patrons.....	20
2.4.1. Exploració dels patrons	22
2.4.2. Exploració dels forms	22
2.4.3. Exploració de les parts d'un form	23
2.4.4. Extracció del requisit	23
2.4.5. Creació de requisits.....	23
2.4.6. Evolució del catàleg	25
2.5. RPToolManager i RPPProjectManager dins del mètode PABRE.....	26
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPPProjectManager.....	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
20. Planificació i Execució del Projecte	245
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager	265

2.1. Introducció

Els patrons de requisits pretenen ajudar a resoldre un dels problemes que habitualment existeixen als llibres de requisits: definicions de requisits ambigües, incompletes, incoherents i normalment expressades de forma poc sistemàtica. L'ús d'aquests patrons permetrà crear una forma estàndard en l'elaboració i redacció de requisits, i també ajudarà a expressar els requisits de forma general i clara, millorant així la qualitat d'aquests, i reduirà el temps i l'esforç de l'elicitació, portant a estalvis econòmics.

Ens centrarem en l'ús de patrons que el grup de recerca GESSI investiga, és a dir, en l'ús d'aquests patrons en l'àmbit de projectes de selecció de components software mitjançant el mètode call-for-tenders.

Així doncs, definirem en què consisteix un catàleg de patrons de requisits (secció 2.2) i l'estructura d'un patró de requisit tal i com el grup l'entén (secció 2.3). A continuació, a la secció 2.4, descriurem el mètode PABRE, que és el mètode que proposen per facilitar l'etapa d'elicitació de requisits basant-se en els patrons, i que té per objectiu estalviar temps i reduir els errors durant aquesta etapa (per més informació sobre l'estructura d'un patró de requisit i el mètode PABRE veure [RMFQ09]). Per últim, a la secció 2.5, explicarem com encaixen les aplicacions *RPToolManager* i *RPPProjectManager* dins d'aquest mètode.

2.2. Catàleg de Patrons de Requisits

El concepte de Catàleg de Patrons de Requisits fa referència al conjunt de tots els patrons de requisits que s'han definit. D'aquí en endavant, quan parlem de catàleg, ens estarem referint a aquest mateix concepte.

2.2.1. Classification schemas

Per facilitar la comprensió i la reutilització dels patrons de requisits durant el procés d'elicitació, els patrons necessiten ser indexats seguint algun esquema de classificació jeràrquic dins del catàleg. En concret, es permetrà tenir més d'un esquema de classificació dels patrons del catàleg. El motiu és millorar tant la usabilitat com la portabilitat del repositori: usabilitat, perquè el mateix catàleg pot ser usat amb diferents esquemes de classificació pel mateix enginyer de requisits; portabilitat, perquè diferents enginyers de requisits, acostumats a altres estàndards o inclús al seu propi, personalitzaran els classification schemas i podran veure el catàleg de patrons de requisits des de la seva pròpia perspectiva.

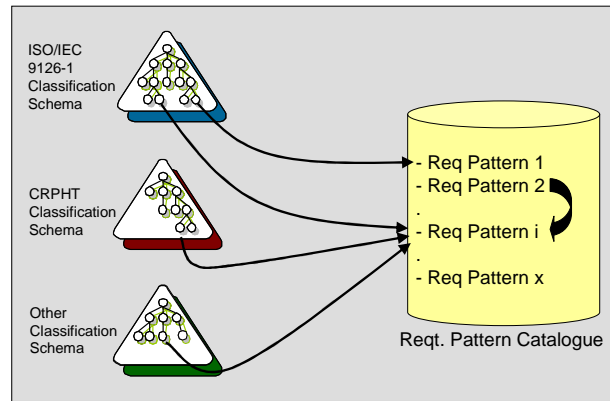


Figura 1: Organització del catàleg de patrons de requisits

2.3. Estructura d'un Patró de Requisit

La plantilla per a un patró de requisit que es proposa a [RMFQ09] es mostra a la Taula 1, omplerta amb un exemple particular, el patró d'*Avís d'errors*.

Requirement Pattern <i>Avís d'errors</i>	Descripció	Aquest patró expressa la necessitat d'una solució software de tenir la capacitat d'informar als usuaris sobre errors.	
	Comentaris	L'avis és llençat en el moment que l'error succeeix.	
	Goal del patró	Avisar als usuaris sobre errors.	
	Autor	Oscar Mendez-Bonilla	
	Fonts (0..*)	<ul style="list-style-type: none"> Llibre de requisits del CITI. Literatura especialitzada. 	
	Paraules clau (0..*)	Avis, Error, Crash	
	Dependències (0..*)	IMPLICA: Informe d'errors	
Requirement Form <i>Proporcionar avis d'error</i>	Descripció	Aquest form no estableix cap relació entre el tipus d'avis i el tipus d'error.	
	Comentaris	Cada extensió pot ser aplicada només un cop.	
	Versió	Dimecres, 26/11/2008 - 2:25am	
	Autor	Oscar Mendez-Bonilla	
	Fonts (0..*)	<ul style="list-style-type: none"> Llibre de requisits del CITI. Literatura especialitzada. 	
	Fixed Part	Question Text	Proporciona la solució avisos en cas d'error?
		Form Text	La solució ha de proporcionar un avis en cas d'error.
	Extended Part <i>Tipus d'avis</i>	Question Text	Permet la solució avisar d'errors als usuaris mitjançant mecanismes concrets? Quins són aquest mecanismes?
		Form Text	Els avisos proporcionats per la solució han de ser: AL
		Paràmetre	Mètrica
		AL: és un conjunt no buit de tipus d'avis	AL: Set(AlertType) AlertType: {E-mail, SMS, Page, Fax, Skype, IM, ...}
	Extended Part <i>Tipus d'error</i>	Question Text	Permet la solució avisar només de tipus d'errors concrets? Quins són aquests tipus?
		Form Text	Els errors que han de ser avisats són: FL
		Paràmetre	Mètrica
		FL: és un conjunt no buit de tipus d'error	FL: Set(FailureType) FailureType: {Server Crash, Network Crash, ...}
Requirement Form <i>Tipus d'avis dependent del tipus d'error</i>	Descripció	Aquest form estableix una dependència entre el tipus d'avis i el tipus d'error que succeeix.	
	Comentaris	Les extensions poden ser aplicades més d'un cop.	
	Versió	Dimecres, 26/11/2008 - 2:45am	
	Autor	Carme Quer	

	Fonts (0..*)	<ul style="list-style-type: none"> • Llibre de requisits del CITI. • Literatura especialitzada. 	
	Fixed Part	Question Text	Proporciona la solució avisos en funció del tipus d'error produït?
		Form Text	La solució ha de donar avis de cert tipus depenent del tipus d'error.
	Extended Part Dependència específica	Question Text	Permet la solució avisar mitjançant mecanismes concrets segons el tipus d'error produït? Quin són els tipus d'avisos en funció del tipus d'error proporcionats per la solució?
		Form Text	Una avis d'un dels tipus AL ha de ser proporcionat per a un error d'algun dels tipus FL.
		Paràmetre	Mètrica
		AL: és un conjunt no buit de tipus d'avis	AL: Set(AlertType) AlertType: {E-mail, SMS, Page, Fax, Skype, IM, ...}
		FL: és un conjunt no buit de tipus d'error	FL: Set(FailureType) FailureType: {Server Crash, Network Crash, ...}

Taula 1: Plantilla i exemple d'un patró de requisit

2.3.1. Meta-dades del patró

El primer conjunt d'atributs defineixen les meta-dades sobre el mateix patró: el nom del patró, la seva descripció, el seu autor, comentaris inclosos per l'autor i els usuaris, el seu goal, les fonts des d'on el patró va ser obtingut (és a dir, els llibres de requisits i projectes des dels quals el patró va ser identificat i inclòs al repositori) i algunes paraules clau per facilitar la cerca al repositori. Cal remarcar aquí el rol important que juga l'atribut goal, ja que ajudarà a decidir si el patró és aplicable al projecte donant-li només una ullada (per més informació veure el punt 2.4): un patró serà afegit al llibre de requisits si el client necessita que el componenet a adquirir tingui com a objectiu el goal del patró en qüestió. En altres paraules, el goal juga el rol de la part del “problema” del patró.

En el cas del patró *Avis d'errors*, el goal és que els usuaris del component que serà solució volen ser avisats quan algun error succeeix.

2.3.2. Requirement forms

La “solució” al problema és encapsulada en els requirement forms del patró. Un patró de requisit, quan és usat en diferents projectes per aconseguir el mateix goal, pot ser expressat de diferents maneres, per això la plantilla permet declarar més d'un form diferent, i per tant més d'una solució, en un mateix patró.

Normalment, el nombre de requirement forms diferents en un patró serà molt baix. Per exemple, el patró *Avis d'errors* té dos requirement form que difereixen en la granularitat de la informació necessitada: si el client necessita algun tipus específic d'alerta quan algun tipus específic d'error succeeix (requirement form *Tipus d'avis depenent del tipus d'error*) o no (*Proporcionar avis d'error*).

Cada form té algunes meta-dades similars a les d'un patró, que són: el nom del requirement form, la seva descripció, el seu autor, comentaris inclosos per l'autor i els usuaris, la seva versió o data en el qual el form ha sigut modificat per últim cop, i les

fonts des d'on ha sigut obtingut. Finalment, cada form té també uns atributs específics, que corresponen a una part fixa, o *fixed part*, que pot tenir una o més extensions.

2.3.3. Fixed part

La *fixed part* d'un requirement form és normalment bastant abstracta: la inclusió en un llibre de requisits del requisit obtingut per l'aplicació d'una *fixed part* assenyala que el sistema ha d'aconseguir el goal del patró, però no com aquest goal és aconseguit. En el cas del requirement form *Proporcionar avís d'error*, la *fixed part* indica que s'informarà de l'error mitjançant avisos.

2.3.4. Extensions

Donat que la *fixed part* d'un requirement form és abstracta, és usual conèixer alguna informació extra o restriccions sobre com aconseguir el goal del patró. Les extensions ("extended part" en la plantilla) permeten assenyalar aquesta informació.

Les extensions poden ser definides o bé reescriuint la *fixed part* o bé restringint aquesta. En el cas del requirement form *Proporcionar avís d'error*, es defineixen extensions per establir el tipus d'avisos que són demanats pel client i el tipus d'errors que necessiten ser informats. L'ús d'extensions, per tant, permet incloure informació més detallada en el llibre de requisits quan s'aplica el patró.

2.3.5. Form text

Cada *fixed* i *extended part* d'un patró és especificada per un form text. Aquest text és expressat com una frase curta escrita en llenguatge natural que pot incloure o no un o més paràmetres que indiquen aquelles parts que poden variar en diferents projectes. En el cas del patró *Avís d'errors*, hi ha paràmetres en les *extended part* dels requirement forms.

Quan un patró és seleccionat i un form aplicat, els paràmetres que apareixen en el text seran substituïts per valors. Per tal de definir els valors vàlids que un paràmetre pot prendre, cada paràmetre tindrà associada una mètrica i opcionalment també tindrà una condició de correctesa. Les mètriques poden ser valors enumerats (per exemple, nom de plataformes middleware), números enters (per indicar, per exemple, el nombre de connexions suportades), números reals (per mesurar el temps de resposta), valors booleans (per saber si un tipus de protocol és o no suportat, per exemple) i time point (per senyalar, per exemple, quan volem que es realitzi un backup de les dades del sistema). En l'exemple, el paràmetre de la primera *extended part* del requirement form *Proporcionar avís d'error*, prendrà com a valor un conjunt no buit de tipus d'avisos que el client desitja que la solució proporcioni.

2.3.6. Question text

Cada fixed i extended part d'un patró té una question text associada. Aquesta question és expressada com una frase curta escrita en llenguatge natural i es correspon amb el que es preguntaria quan es vol saber si un producte compleix o no el requisit marcat per la fixed o extended part a la que pertany. En el cas de la primera extended part del requirement form *Proporcionar avis d'error*, la question text és: "Permet la solució avisar d'errors als usuaris mitjançant mecanismes concrets? Quins són aquests mecanismes?".

2.3.7. Dependències

Els patrons no viuen aïllats i poden estar interrelacionats amb el catàleg. S'han identificat dos tipus de dependències, entre patrons i entre paràmetres.

Les dependències entre patrons generalitzen la idea de tenir dependències entre requisits. Aquestes dependències poden ser usades durant el procés d'elicitació (per ajudar a determinar l'ordre d'aplicació entre patrons) i també poden ser propagades a l'especificació de requisits per millorar la traçabilitat (si el catàleg de patrons ha informat que aconseguir un requisit influencia en aconseguir o no un altre).

Per l'altre costat, tot i que no apareixen en la plantilla i encara no han sigut cas d'estudi, es van identificar també dependències entre paràmetres. Aquestes dependències poden ajudar a simplificar el procés i reforçar la correctesa del llibre de requisits resultant, ja que aquestes dependències declararan relacions entre els valors dels diferents paràmetres que han de ser omplerts. Per exemple, donat el patró de requisit "El manual d'usuaris ha de ser escrit en <nom_de_l'idioma_del_manual>" i "El servei d'ajuda de l'aplicació donarà assistència en <nom_de_l'idioma_de_l'ajuda>", tenint en compte que ambdós paràmetres han sigut declarats de tipus *IdiomaOficial* (una mètrica enumerada amb valors {Anglès, Castellà, Català, ...}), la relació pot senyalar que el valor per defecte del segon paràmetre sigui el mateix idioma que l'escollit al primer.

2.4. Mètode PABRE: Creació i utilització de patrons

El mètode PABRE és el mètode proposat a [RMFQ09] per explorar el catàleg de patrons de requisits per produir un llibre de requisits i per generar un document per ser usat en un procés call-for-tenders. El mètode facilita l'etapa d'elicitació de requisits basant-se en els patrons, amb l'objectiu d'estalviar temps i reduir els errors durant aquesta activitat. A la Figura 2 es pot observar quines són les fases d'aquest mètode.

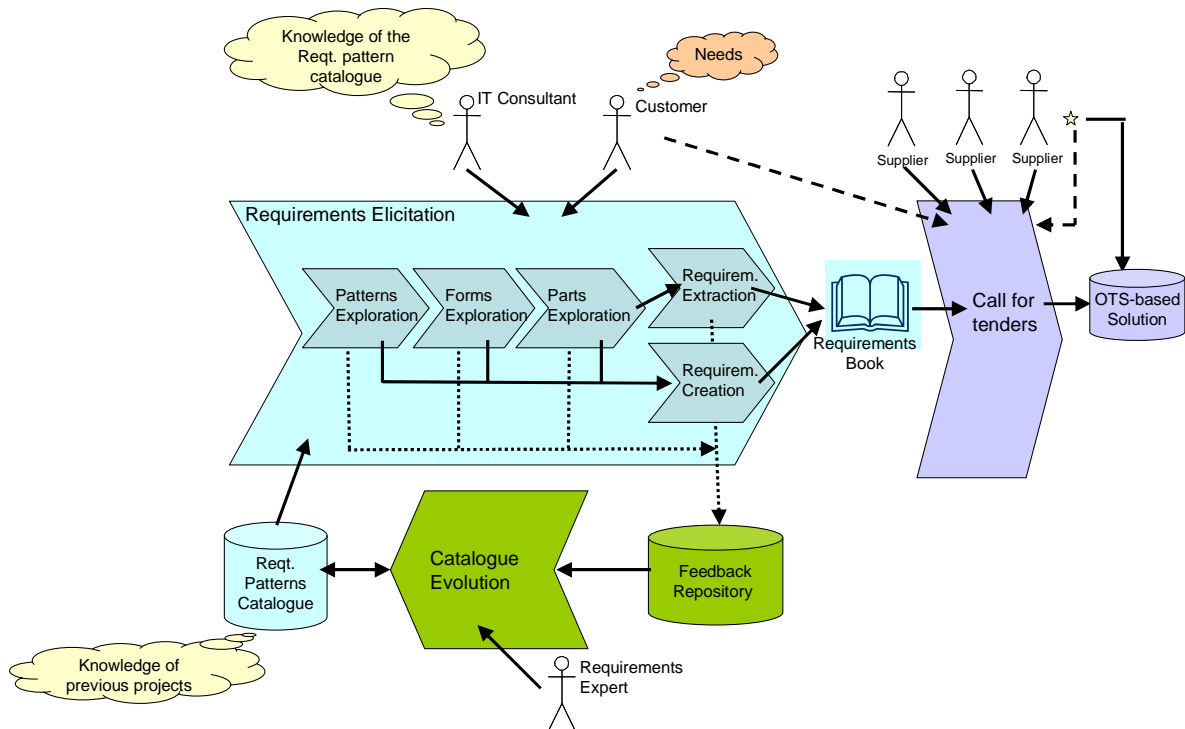


Figura 2: Mètode PABRE, procés basat en patrons per a la selecció d'una solució

Cal tenir en compte, però, que tot i que és esperat que la gran part dels requisits que formen part del llibre de requisits resultants vinguin directament de la instanciació de patrons, altres situacions es poden produir:

- 1) Un patró ha de ser lleugerament modificat quan es converteix en un requisit, probablement perquè alguns forms o extensions falten al patró.
- 2) Algun requisit pot no ser creat com una instanciació d'un patró, o bé perquè el requisit és molt específic del projecte o bé perquè el catàleg encara no està del tot complet.

L'elicitació de requisits és, seguint aquest mètode, un procés de cerca i selecció en el catàleg de patrons de requisits. En el nostre cas serà realitzat a través d'entrevistes entre el consultant IT i el client (o bé un membre representatiu del client). Totes les decisions seran preses amb l'acord entre el consultant IT i el client.

Abans de començar l'exploració dels patrons, el consultant IT ha d'escollir un classification schema que guiarà la seva exploració del catàleg. Normalment, el consultant IT seleccionarà el classification schema que li sigui més familiar.

A l'inici del procés, el consultant IT explica al client el procediment que serà seguit. Més concretament, el client ha de rebre de forma molt clara dos missatges:

- 1) El flux de requisits: els patrons seran explorats d'acord amb el classification schema seleccionat: "requisits estandaritzats" (és a dir, els patrons) seran proposats per un àmbit de requisits donats (és a dir, un conjunt de classificadors relacionats) i quan tots els "requisits estandaritzats" d'un àmbit concret han sigut explorats, llavors l'àmbit canviarà i nous "requisits estandaritzats" seran proposats.
- 2) Processament individual: per cada patró hi ha un seqüència definida de passos que es portaran a terme de forma sistemàtica.

A continuació es presenten els diferents passos del procés, en el qual hi ha clarament dos fases diferenciades: l'elicitació de requisits per si mateixa, la qual consisteix en cinc passos aplicats iterativament, i la fase d'evolució del catàleg, la qual manté el coneixement relacionat amb patrons actualitzat.

2.4.1. Exploració dels patrons

A cada iteració, el consultant IT comença seleccionant el següent patró aplicable d'acord amb l'actual criteri de classificació (pas S.0 de la Figura 3), i un cop comprovats els comentaris sobre el patró, si és que n'hi ha, explica la descripció i goal del patró al client (S.1). Basant-se en l'explicació, el consultant IT junt amb el client defineixen la importància del patró (decisió D.1 de la Figura 3). Si el client considera que el patró no és important pel seu projecte, el consultant IT determina amb el client si el patró satisfà les necessitats del client (D.2). Aquesta decisió permetrà de manera ràpida descartar el patró sense haver-lo de processar sencer (S.2).

Descartar patrons en aquesta fase és només permès quan el client considera que el goal del patró no iguala cap de les seves necessitats (marcant la importància com "low") . Quan els patrons són descartats, el consultant IT recull el motiu per poder realitzar futurs anàlisis qualitatius del catàleg. Llavors el consultant IT continua amb el següent patró d'acord amb la classificació.

Si el patró seleccionat en aquesta iteració és l'últim de l'àmbit actual, abans de canviar d'àmbit, el consultant IT preguntarà al client si hi ha encara alguna necessitat relacionada amb aquest àmbit que no hagi estat coberta amb els patrons (D.3). Si aquest és el cas, és necessari crear un o més requisits des de zero (veure secció 2.4.5 Creació de Requisits).

2.4.2. Exploració dels forms

Si el client selecciona el patró, això significa que un o varis requisits que s'assemblin al goal del patró han d'aparèixer al llibre de requisits. El consultant IT ha d'explicar els diferents forms del patró, basant-se en la seva descripció (S.3). El client llavors escull el form més apropiat per aconseguir el goal del patró d'acord al seu context (S.4). La descripció del form escollit pot ja ser considerada com una part del requisit que serà inclòs al llibre de requisits. Si no existeixen forms que s'adaptin a les necessitats del client, el consultant IT necessitarà elaborar el requisit per tal de satisfer el goal del patró; aquesta informació també serà considerada com feedback per a la fase d'evolució del catàleg.

2.4.3. Exploració de les parts d'un form

Si algun form ha sigut seleccionat, el consultant IT explica les diferent parts que componen el requirement form escollit (S.5). El consultant explica breument cada paràmetre i dona exemples de possibles valors (trobat a la descripció de la mètrica associada al paràmetre). El client escull la part més convenient per aconseguir el goal del patró, d'acord amb el seu context (S.6). El client pot escollir més d'una extended part per un form específic; inclús una extended part pot ser aplicada més d'un cop amb diferents assignacions als valors dels paràmetres (per un exemple, veure la plantilla de patró de la Taula 1 anterior). Eventualment, alguna extensió no existent en el catàleg pot ser necessitada, i una altre cop torna a ser necessari definir les necessitats que falten de forma separada, com a un nou requisit no associat a cap form existent al catàleg.

2.4.4. Extracció del requisit

Per les parts escollides, el consultant IT dóna més detalls sobre els paràmetres que aplicar, és a dir, detalls sobre les possibles condicions de correctesa, dependències a/entre paràmetres (si aquestes fossin tractades) i explica la llista exhaustiva de valors per cada paràmetre (S.7). A continuació el client escull els valors pels paràmetres (S.8). El requisit és extret aplicant el form text de les parts seleccionades amb els valors dels paràmetres que s'han acordat entre el client i el consultant IT. Diferents form texts de diferents extensions poden ser concatenats per extreure un únic requisit. Per exemple, usant la plantilla de la Taula 1, un possible requisit que provindria del primer form i d'aplicar les seves dues extensions un cop, podria ser: "La solució ha de proporcionar un avís en cas d'error. Els avisos proporcionats per la solució han de ser: E-mail, SMS. Els errors que han de ser avisats són: ServerCrash, NetworkCrash."

Durant la selecció dels valors, les relacions de dependències entre patrons han de ser comprovades (D.4). Els diferents tipus de dependències s'han de tenir en compte (conflicte, sinèrgies, etc.) així com la seva direcció (des de quin patró a quin altre, és a dir, la dependència pot ser a un patró ja tractat o a un patró encara no considerat durant el procés). Quan el consultant IT detecta el conflicte o una inconsistència, ha d'avisar al client i s'ha d'intentar solucionar el conflicte (S.9). La resolució del conflicte pot no ser senzilla i fins i tot pot forçar a reconsiderar requisits ja acordats; per ara, però, aquesta qüestió encara no es té en compte.

2.4.5. Creació de requisits

Després d'explicar les fases anteriors, queda clar que en diferents situacions l'extracció del requisit des del catàleg pot no ser directa. S'han identificat les següents situacions:

- Els patrons que pertanyen a un mateix àmbit no cobreixen totes les necessitats del client relacionades amb aquest àmbit. El consultant IT utilitza la informació sobre els criteris de classificació de l'àmbit per guiar un clàssic procés d'elicitació de requisits (S.10).

- Un patró s'ha considerat aplicable però cap dels seus forms encaixa bé amb les necessitats del client. El consultant IT treballarà amb el client fins que es trobi una bona manera d'expressar el goal del patró que encaixi amb les necessitats del client (S.11).
- Un requirement form ha sigut escollit però alguns detalls necessaris no són expressats en cap extension existent. El consultant IT treballarà amb el client tots els detalls necessaris per completar el requisit amb tota la informació rellevant (S.12).

En tots els casos, el requisit (o part d'ell) afegit al llibre de requisits és proporcionat com a feedback a l'expert en requisits per a la fase següent de l'evolució del catàleg.

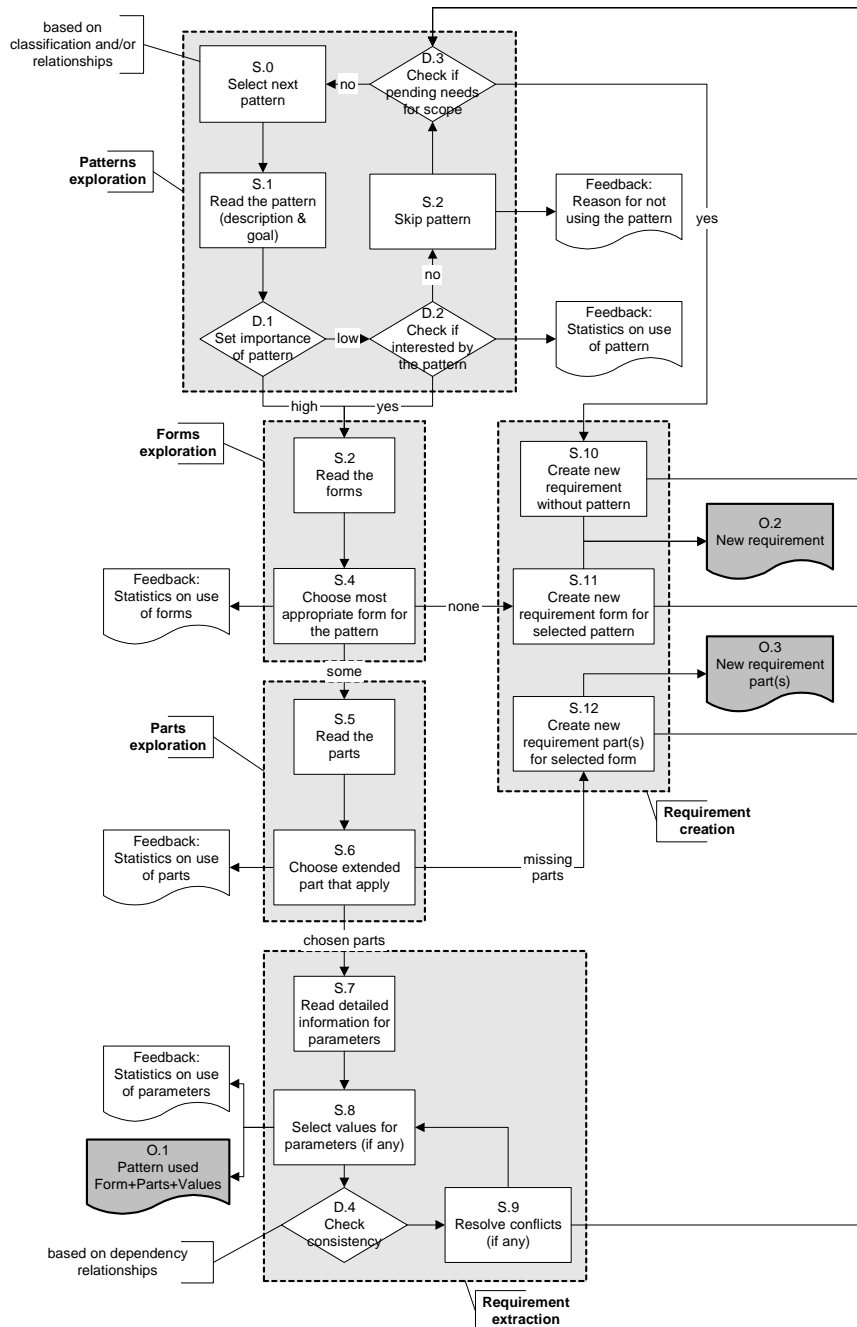


Figura 3: Detall de les activitats que es porten a terme durant la fase d'elicitació de requisits

2.4.6. Evolució del catàleg

Després de que el consultant IT ha dirigit el procés d'elicitació de requisits i el llibre de requisits pel call-for-tenders és completat, el coneixement adquirit en aquest projecte ha de ser capitalitzat en el catàleg de patrons de requisits. Com s'ha senyalat a les fases anteriors, el consultant IT recollirà la informació útil per aquest propòsit, tant fallades com encerts en l'aplicació de patrons.

Per les fallades, l'apartat anterior de Creació de Requisits resumeix les situacions que poden ser trobades en les quals el catàleg no ha disposat de tota la informació per aquest projecte. Per als encerts, cada aplicació de patró és registrada. Cal tenir en compte, però, que també pot haver-hi casos al mig; per exemple, quan un requisit és aplicat però cap form captura el seu goal amb els termes apropiats: això és un èxit des del punt de vista del patró (és a dir, el patró és escollit i aplicat), però una fallada des del punt de vista del form (perquè falta un form).

Les diferents accions que un enginyer de requisits pot emprendre per ampliar el catàleg, i les situacions en les quals aquestes seran portades a terme, són:

- Extreure d'un requisit un patró de requisit. Quan un requisit és escrit des de zero (veure el primer punt de l'apartat 2.4.5 Creació de Requisits).
- Crear un nou requirement form. Quan un requisit prové del goal d'un patró però de cap form existent (veure el segon punt de l'apartat 2.4.5 Creació de Requisits).
- Crear una nova extension. Quan un requisit és expressat usant un form però els detalls necessaris no són continguts en cap extension existent (veure el tercer punt de l'apartat 2.4.5 Creació de Requisits).
- Estendre alguna extension existent. Quan una extension conté alguna mètrica però les restriccions d'aquesta no s'adapten al projecte.

Però no només és possible fer més gran el catàleg, sinó que algunes altres operacions poden ser aplicades sobre el catàleg després d'actualitzar les estadístiques d'ús amb les dades d'aquest projecte:

- Eliminar patrons, forms o extensions no usats. Quan després d'aquest projecte algun threshold s'ha excedit, podria ser considerada l'eliminació del catàleg d'aquelles peces d'informació que no semblen ser rellevants en un procés call-for-tenders.
- Refactoritzar el catàleg. Per exemple, canviant l'ordre en el qual les forms d'un patró són considerades (les més usades les primeres, per exemple), o inclús canvis fonamentals com separar o unir algun patró.

La decisió de portar a terme o no aquestes accions serà de l'expert en requisits, probablement consultant abans de prendre la decisió amb el consultant IT.

2.5. *RPToolManager* i *RPProjectManager* dins del mètode PABRE

Primer de tot, explicarem quines són, a grans trets, les utilitats de les aplicacions *RPToolManager* i *RPProjectManager*:

- L'aplicació *RPToolManager* és una aplicació per a la definició i modificació de patrons de requisits i la gestió d'un repositori per aquests.
- L'aplicació *RPProjectManager* s'utilitza per a instanciar els patrons de requisits en projectes de selecció i poder generar el document necessari pel call-for-tenders. Aquesta aplicació permet gestionar projectes de selecció, navegar pel repositori de patrons segons diferents criteris, instanciar patrons de requisits que es troben en el repositori i generar finalment el document de call-for-tenders.

Així doncs, tenint en compte quines són les funcionalitats de cadascuna d'aquestes aplicacions, l'esquema del mètode PABRE mostrat a la Figura 2 anterior, es veuria modificat tal i com es mostra a la Figura 4 següent.

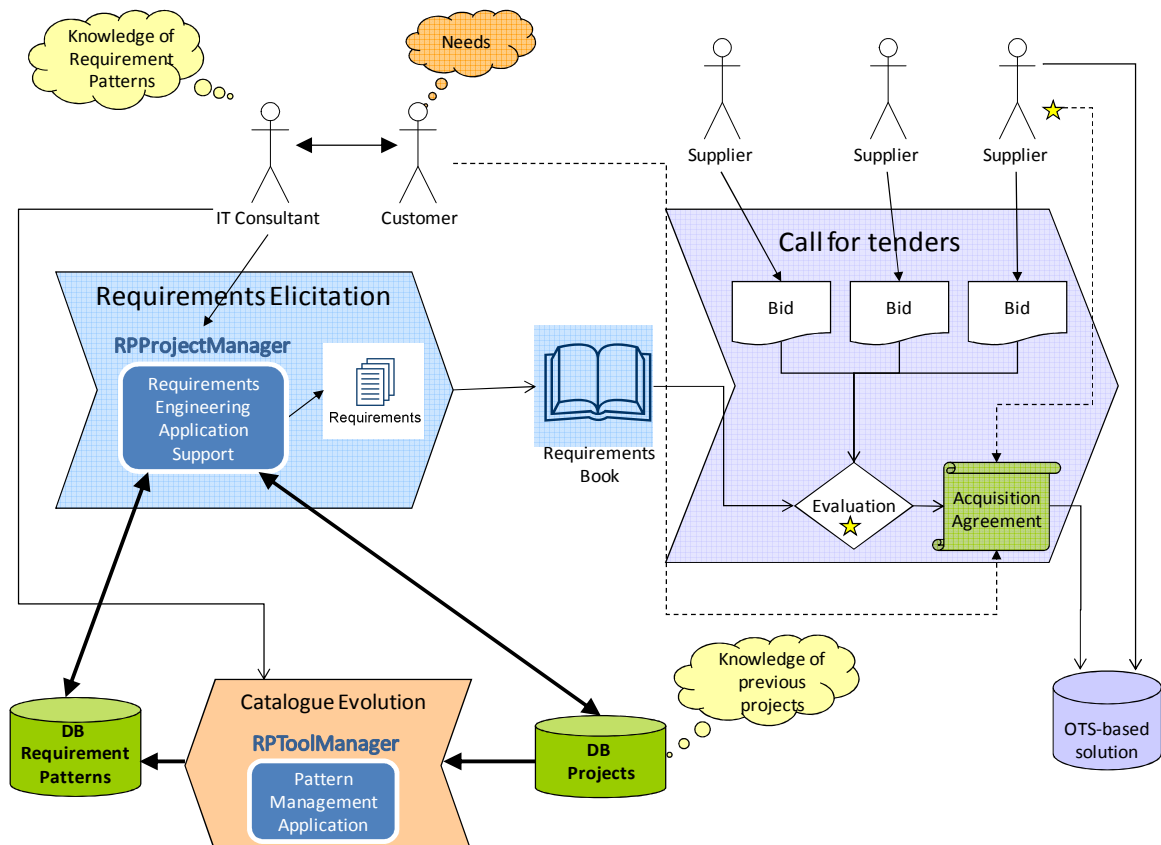


Figura 4: Mètode PABRE amb les aplicacions *RPToolManager* i *RPProjectManager*

Inicialment, com ja s'ha explicat a apartats anteriors, hi haurà un client que té una necessitat que es pot resoldre mitjançant un component software. A partir d'aquest moment començarà la fase d'elicitació i definició dels requisits que ha de complir aquest component: mitjançant el feedback amb el client, el consultor IT coneixerà quines són les necessitats d'aquest i, amb l'ajuda de l'aplicació *RPPProjectManager*, podrà instanciar a partir dels patrons els diferents requisits (o bé en els casos que no hi hagi patrons definits, crear nous requisits), obtenint així els requisits que passaran posteriorment al document de call-for-tenders. Un cop generat aquest document, que a la Figura 4 s'anomena *Requirements Book*, només queda fer-lo arribar als diferents proveïdors, els quals faran les seves propostes del producte o productes que poden complir amb els requisits exposats. Posteriorment aquestes propostes s'avaluaran, seleccionant així els productes que satisfan les necessitats del client.

Per altra banda, en la fase d'evolució del catàleg, partint del coneixement que s'ha obtingut en cada nou projecte (patrons que s'utilitzen molt poc, requisits nous que s'han afegit a un llibre de requisits però que no són instància de cap patró, o fins i tot que sí que són instància d'un patró però o bé cap de les seves forms o bé cap valor dels paràmetres s'adaptava a les necessitats del client), el consultor IT o bé un expert en requisits podrà actualitzar el repositori de patrons amb el nou coneixement adquirit mitjançant l'aplicació *RPToolManager*.

Cal tenir en compte, però, que ara per ara les versions desenvolupades en aquest projecte tant de l'aplicació *RPToolManager* com de *RPPProjectManager* encara no assoleixen totes les funcionalitats necessàries per cobrir el procés tal i com s'ha explicat en aquest apartat. Per més informació sobre l'abast del projecte i el treball que queda per fer, consultar els capítols 11.2 i 21.

3. Eines de Treball

1.	Introducció.....	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
3.1.	<i>Tecnologies utilitzades</i>	<i>30</i>
3.1.1.	Java: Llenguatge de Programació	30
3.1.2.	Hibernate: del SGBD Relacional a l'Orientació a Objectes	31
3.2.	<i>Entorn de Desenvolupament (IDE)</i>	<i>32</i>
3.3.	<i>SQirreL-SQL Client: Vista i modificació d'una Base de Dades existent.....</i>	<i>33</i>
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades.....	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager		79
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work.....	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD.....	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

3.1. Tecnologies utilitzades

Durant el transcurs de tot el projecte s'han utilitzat diverses eines i tecnologies. En aquest capítol hi ha una petita explicació de totes elles. Concretament, es vol mostrar com són i com funcionen aquelles parts d'aquests components tecnològics que s'han utilitzat per desenvolupar la 2^a versió de l'aplicació *RPToolManager* i la 1^a versió de l'aplicació *RPPProjectManager*.

3.1.1. Java: Llenguatge de Programació

Motivació

Per a la codificació de les dues aplicacions s'ha utilitzat el llenguatge orientat a objectes Java.

El principal motiu és que aquest va ser el llenguatge escollit per Oriol Arcas per construir la versió 1 de l'aplicació *RPToolManager*, i portar l'eina a un nou llenguatge seria per una banda molt costós i per l'altre contraproductiu, ja que els esforços dedicats a portar l'eina a un llenguatge es podrien dedicar a millorar les funcionalitats de l'aplicació. A més, de cara a la reutilització de codi de l'aplicació *RPToolManager*, sobretot en la part gràfica, per a l'aplicació *RPPProjectManager* ha fet que el llenguatge de programació per aquesta última també sigui Java. Per altra banda, el fet d'utilitzar Hibernate per realitzar el mapeig de les bases de dades relacionals que hi ha al darrere per als objectes de les aplicacions, obliga a que el llenguatge de programació escollit sigui aquest.

Per altra banda, la decisió d'usar Java en la primera versió de l'aplicació es continua considerant encertada donades les característiques i avantatges d'aquest llenguatge de programació [WEB Java]:

- És un llenguatge d'Alt Nivell perfectament Orientat a Objectes
- Existeix una àmplia llibreria d'APIS integrats en la versió oficial de Java de Sun Microsystems, àmpliament testejats, robustos i molt ben documentats, que faciliten la feina al programador.
- La portabilitat és una de les grans avantatges i emblemes del llenguatge de programació, fet que permet seguir utilitzant la mateixa eina en el futur, malgrat canvis de sistema operatiu.
- Una de les característiques negatives de Java és que, al ser un llenguatge interpretat, hi ha una gran pèrdua d'eficiència, però en el cas de les aplicacions que es desenvolupen, per les operacions que es tracten, no és necessari l'ús de cap algorisme altament costós i l'eficiència no és un punt clau del projecte.

Versió

S'ha treballat amb Java 2 Standard Edition (J2SE) i el Java Development Kit (JDK), de la versió oficial 6.0 de Sun Microsystems. No es garanteix el correcte funcionament de les eines amb versions anteriors a aquesta.

Llicència

A partir del 13 de Novembre del 2006, Sun va publicar la majoria del codi de Java com a software lliure sota els termes de GNU General Public License (GPL). A partir del 8 de Maig del 2007, Sun va completar el procés fent que tot el nucli del codi de Java fos codi font obert, exceptuant petites porcions de codi en les que Sun no té els drets de copyright.

3.1.2. Hibernate: del SGBD Relacional a l'Orientació a Objectes

Hibernate Overview

Hibernate és una eina de mapeig objecte-relacional per a la plataforma Java, que facilita el mapeig d'atributs entre una base de dades relacional tradicional i el model d'objectes d'una aplicació, mitjançant arxius declaratius (xml) que permeten establir aquestes relacions.

Hibernate, com d'altres eines amb el mateix propòsit, intenta solucionar el problema de la diferència entre els dos models de dades coexistents en una aplicació: l'utilitzat a la memòria de l'ordinador (orientat a objectes) i l'utilitzat en la base de dades (model relacional). Per aconseguir-ho, permet al programador definir com és el seu model de dades, quines relacions existeixen i quina forma tenen mitjançant els arxius .hbm (arxius amb extensió Hibernate Mapping, que no són més que fitxers declaratius en xml). Amb aquesta informació, Hibernate permet a l'aplicació manipular les dades de la base de dades com si fossin objectes, amb totes les característiques de la programació orientada a objectes. Així doncs, Hibernate convertirà les dades entre els tipus utilitzats per Java i els definits per SQL.

A més, Hibernate està dissenyat per ser flexible en quant a l'esquema de taules utilitzat: pot adaptar-se el seu ús a una base de dades ja existent o, en altre cas, disposa de la funcionalitat de crear la base de dades a partir de la informació disponible en l'arxiu que defineix el mapeig.

A diferència d'altres solucions de persistència, Hibernate no amaga la potència del llenguatge SQL ja que permet expressar queries a la base de dades de tres maneres diferents:

1. Amb el llenguatge natiu SQL.
2. Amb HQL: l'extensió d'Hibernate de SQL.
3. Amb Criteria's orientats a objectes: sentències que treballen directament amb les classes dels objectes.

Motivacions

El principal motiu de la utilització d'Hibernate per al mapeig de la base de dades és que aquesta va ser l'eina escollida per construir la versió 1 de l'aplicació *RPToolManager*, i canviar la capa de dades de l'aplicació per utilitzar una altra eina seria molt costós. Per altra banda, el fet d'utilitzar Hibernate en l'aplicació *RPPProjectManager* s'ha degut al fet que un cop et familiaritzes amb l'eina i els fitxers de mapeig, utilitzar Hibernate suposa un esforç mínim de cara al programador: és molt simple, i gairebé intuïtiu, voler aconseguir tots els objectes d'una taula o un en concret, guardar un objecte que hi ha a la capa de domini, actualitzar-lo a la base de dades o esborrar-lo. A més, les característiques i avantatges d'Hibernate fan que sigui l'eina més útil per a les nostres aplicacions [WEB Hibernate]:

- Té un model de programació natural, és a dir, suporta l'herència, el polimorfisme, la composició i els frameworks de col·lecció de Java.
- Té suport per a un model d'objectes molt fi, és a dir, té una gran varietat de mapeig per a col·leccions i objectes dependents i fan que gairebé tota relació possible en Java es pugui mapejar sense problemes.
- La seva utilització no requereix temps extra durant la construcció/execució de l'aplicació que l'utilitza.
- Ofereix múltiples formes de fer queries a la base de dades i permet no solament aconseguir objectes de la base de dades, sinó també posar-los en ella.
- Té suport per a “conversacions”, és a dir, suporta contexts de llarga vida de persistència dels objectes, fent un locking òptim d'aquests de manera automàtica.

Versió

S'ha treballat amb la distribució oficial d'Hibernate 3.3.2 de Red Hat. No es garanteix el correcte funcionament de les eines amb versions anteriors a aquesta.

Llicència

La llicència d'Hibernate és LGPL (GNU Lesser General Public License). La LGPL és suficientment flexible com per a permetre usar Hibernate tant en projectes open source com comercials.

3.2. Entorn de Desenvolupament (IDE)

L'Eclipse és un conjunt de biblioteques per a crear entorns de desenvolupament integrat. El que es coneix habitualment, però, és l'Eclipse SDK, un entorn de desenvolupament integrat que inclou eines per al desenvolupament d'aplicacions en

Java: Java Development Tools, Eclipse Java Compiler, integració amb CVS, etc. [WEB Eclipse]

Va ser creat per IBM com a successor del VisualAge, i actualment és desenvolupat per l'Eclipse Foundation, una organització sense afany de lucre.

L'arquitectura d'Eclipse està pensada per ser completament extensible mitjançant connectors escrits en Java o altres llenguatges de programació.

La base d'Eclipse és la Rich Client Platform, composta per:

- Core platform: arranca l'Eclipse i executa els connectors.
- OSGi: implementa un model de components dinàmic.
- Standard Widget Toolkit: biblioteca d'interfície gràfica.
- JFace: patró model-vista-controlador i altres utilitats per al Standard Widget Toolkit.
- Eclipse Workbench: implementa les vistes, editors, perspectives i auxiliars.

Llicència

L'Eclipse és distribuït amb una llicència de programari lliure. La seva llicència és EPL (Eclipse Public License) i obliga a distribuir qualsevol producte construït sobre Eclipse o amb parts d'aquest sota la mateixa llicència. En el cas de *RPToolManager* i *RPPProjectManager* la llicència és totalment adequada.

3.3. Squirrel-SQL Client: Vista i modificació d'una Base de Dades existent

L'Squirrel-SQL Client [WEB Squirrel] és una aplicació Java que proporciona una interfície gràfica simple per a bases de dades relacionals. A l'estar escrita en Java, pot accedir a qualsevol base de dades que segueixi JDBC (Java DataBase Connectivity) executant-se en qualsevol màquina, permetent l'accés remot a múltiples bases de dades.

Un usuari d'aquesta aplicació pot, entre d'altres:

- Veure i editar dades fàcilment en qualsevol base de dades que segueixi JDBC.
- Veure les meta-dades de la base de dades.
- Treballar amb múltiples bases de dades, tant en màquines remotes com locals.

- Usar una única interfície per treballar amb diferents motors de bases de dades.
- Expandir les capacitats de l'eina i incloure funcionalitats específiques a la base de dades usant plugins.

L'usuari pot clicar en les taules per veure-les i editar les seves dades, o bé usar directament sentències SQL. Les dades poden ser vistes en mode read-only per seguretat, o en mode editable permetent modificar les dades simplement escrivint les noves dades dins de la taula. Totes les meta-dades de la base de dades (tipus de les dades, nom de les columnes de les taules, etc.) són accessibles a través de Squirrel.

En casos on múltiples tipus de motors de bases de dades estan sent usats (Oracle, MySQL, PostgreSQL, etc.), Squirrel-SQL proporciona un mecanisme comú per accedir a ells. En els casos on un motor de base de dades no segueix cap estàndard, l'arquitectura de plugins de SQL permet a l'usuari incloure components específics per dirigir aquestes operacions per a la seva base de dades.

En el cas de l'aplicació *RPToolManager*, l'Squirrel-SQL ens servirà per afegir noves columnes a taules ja existents de la base de dades que hi ha darrera de l'aplicació d'una manera senzilla i còmoda, i sobretot sense perdre les dades ja introduïdes a la base de dades.

Llicència

La llicència de l'Squirrel-SQL Client és LGPL (GNU Lesser General Public License).

4. Arquitectura: En tres capes

1. Introducció.....	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
4.1. Introducció	36
4.2. Capes i nivells.....	36
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager.	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

4.1. Introducció

La programació per capes és un estil de programació en el qual l'objectiu primordial és la separació de la lògica de negocis de la lògica de disseny; un exemple bàsic d'això consisteix a separar la capa de dades de la capa de presentació a l'usuari.

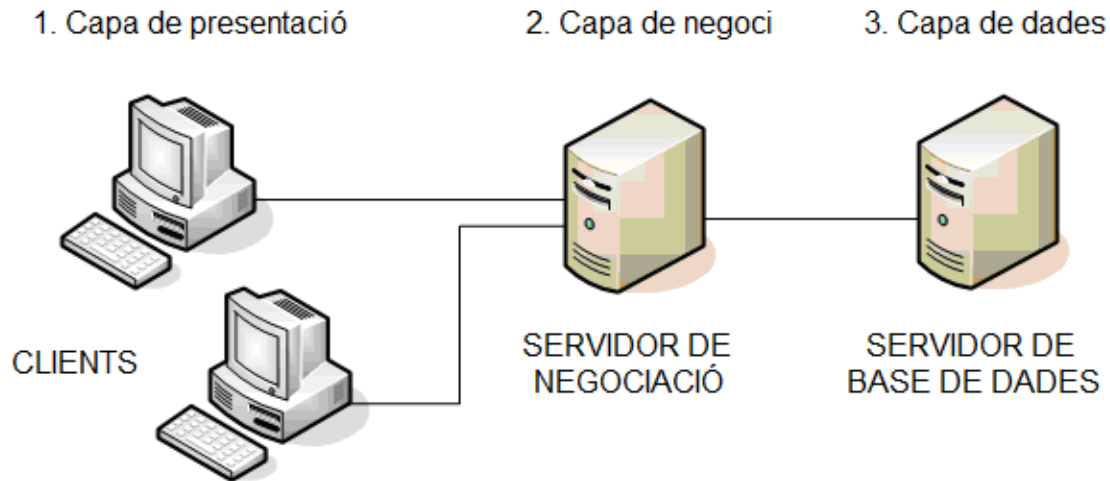


Figura 5: Arquitectura en tres capes

L'avantatge principal d'aquest estil és que el desenvolupament es pot portar a terme en diversos nivells i, en cas que sobrevingui algun canvi, només s'ataca al nivell requerit sense haver de revisar entre codi barrejat.

A més, permet distribuir el treball de creació d'una aplicació per nivells; d'aquesta manera, cada grup de treball està totalment abstret de la resta de nivells, de manera que n'hi ha prou amb conèixer l'API que existeix entre nivells.

En el disseny de sistemes informàtics actual se sol usar les arquitectures multinivell o programació per capes. En aquestes arquitectures a cada nivell se li confia una missió simple, el que permet el disseny d'arquitectures escalables (que poden ampliar-se amb facilitat en cas que les necessitats augmentin).

El disseny més utilitzat actualment és el disseny en tres nivells (o en tres capes).

4.2. Capes i nivells

Capa de presentació

És la que veu l'usuari (també se la denomina "capa d'usuari"). Presenta el sistema a l'usuari, li comunica la informació i captura la informació de l'usuari (realitza un filtrat previ per a comprovar que no hi ha errors de format). Aquesta capa es comunica únicament amb la capa de negoci. També és coneguda com interfície gràfica i ha de tenir la característica de ser "amigable" (fàcil d'entendre i d'usar) per a l'usuari.

Capa de negoci o domini

És on resideixen els programes que s'executen, es reben les peticions de l'usuari i s'envien les respostes després del procés. Es denomina capa de negoci (i fins i tot capa de domini o de lògica del negoci) perquè és aquí on s'estableixen totes les regles que han de complir-se. Aquesta capa es comunica amb la capa de presentació, per a rebre les sol·licituds i presentar els resultats, i amb la capa de dades, per a sol·licitar al gestor de base de dades per a emmagatzemar o recuperar dades d'ell. També es consideren aquí els programes d'aplicació.

Capa de dades

És on resideixen les dades i és l'encarregada d'accedir a aquestes. Està formada per un o més gestors de bases de dades que realitzen tot l'emmagatzematge de dades, rebent sol·licituds d'emmagatzematge o recuperació d'informació des de la capa de negoci.

Nivells

En una arquitectura, els termes "capes" i "nivells" no signifiquen el mateix ni són similars. El terme "capa" fa referència a la forma com una solució és segmentada des del punt de vista lògic: Presentació / Lògica de Negoci / Dades. En canvi, el terme "nivell" correspon a la forma que les capes lògiques es troben distribuïdes de forma física. Per exemple:

- Una solució de tres capes (presentació, lògica del negoci, dades) que resideixen en un sol ordinador (presentació+lògica+dades). Es diu que l'arquitectura de la solució és de tres capes i un nivell.
- Una solució de tres capes (presentació, lògica del negoci, dades) que resideixen en dos ordinadors (presentació+lògica, lògica+dades). Es diu que l'arquitectura de la solució és de tres capes i dos nivells.
- Una solució de tres capes (presentació, lògica del negoci, dades) que resideixen en tres ordinadors (presentació, lògica, dades). L'arquitectura que la defineix és: solució de tres capes i tres nivells.

PART 1:
Desenvolupament
de la 2^a versió
de l'aplicació
RPToolManager

5. Manteniment i Extensió de Software

1.	Introducció	11
2.	Patrons de requisits	15
3.	Eines de Treball	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
5.1.	<i>Procés de manteniment i extensió de software</i>	<i>42</i>
5.1.1.	Processos definits en manteniment i extensió de software	42
5.1.2.	Processos definits a l'aplicació RPToolManager	43
5.2.	<i>Tipus de manteniment de software</i>	<i>44</i>
5.2.1.	Corrective Maintenance	45
5.2.2.	Adaptive Maintenance	45
5.2.3.	Perfective Maintenance	45
5.2.4.	Preventive Maintenance	45
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1^a versió de l'aplic. RPProjectManager		79
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades	195
18.	Document de Disseny	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

5.1. Procés de manteniment i extensió de software

En l'enginyeria del software, s'entén per manteniment del software la modificació del producte després del seu lliurament per corregir errors, millorar les prestacions o altres atributs o per adaptar el producte a un entorn diferent a l'inicial (ISO/IEC 14764 [WEB ISO 14764]).

5.1.1. Processos definits en manteniment i extensió de software

Aquest estàndard internacional descriu els sis processos de manteniment del software com:

- 1) El procés d'implementació conté la preparació del software i activitats de transició, com per exemple la concepció i creació d'una pla de manteniment, la preparació del tractament de problemes identificats durant el desenvolupament, i el seguiment en la gestió de la configuració.
- 2) El procés d'anàlisi del problema i modificació, que es fa un cop l'aplicació ha esdevingut responsabilitat del grup de manteniment. El programador de manteniment ha d'analitzar cada requeriment, confirmar-lo (mitjançant la reproducció de la situació) i comprovar la seva validesa, investigar i proposar una solució, documentar la petició i la solució proposada, i, finalment, obtenir totes les autoritzacions requerides per tal d'aplicar les modificacions.
- 3) El procés considerant la implementació una modificació en si mateixa.
- 4) El procés d'acceptació de la modificació, mitjançant la seva comprovació amb els stakeholders implicats en la funcionalitat que havien sol·licitat les modificacions, per tal d'assegurar que el resultat final satisfà els seus requeriments.
- 5) El procés de migració és excepcional, i no és part de tasques de manteniment diàries. Si el software ha de ser portat a alguna altra plataforma sense cap canvi de funcionalitat, el procés serà usat i mantingut per un equip de projecte.
- 6) El darrer procés és el que correspon a la retirada del software un cop és obsolet degut a la nova versió existent del software.

A la Figura 6 es pot observar quin és el cicle dels processos definits en el manteniment i extensió del software.

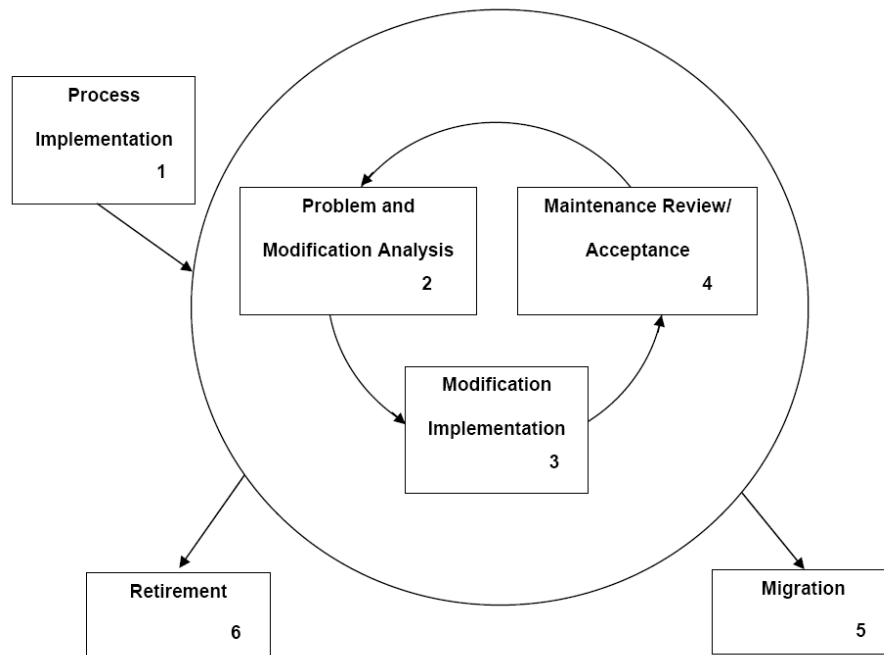


Figura 6: Cicle dels processos de manteniment i extensió del software

5.1.2. Processos definits a l'aplicació RPToolManager

El desenvolupament de la segona versió de l'aplicació *RPToolManager* comportarà només els processos de l'1 al 4 abans esmentats, els detalls dels quals consistiran en el següent:

Procés 1

El procés 1 correspondria en el cas de l'aplicació de *RPToolManager* a tot allò desenvolupat pels mateixos desenvolupadors de la versió 1 de l'aplicació, en aquest cas, l'Oriol Arcas i l'Oriol Colell.

Procés 2

Aquest procés va ser desenvolupat inicialment per l'Oriol Colell mitjançant una documentació d'algunes de les mancances i millores necessàries de l'aplicació. Posteriorment, en un anàlisi de major profunditat del mateix projecte realitzat per mi, es van obtenir noves mancances i millores a desenvolupar. A més, junt amb la client de l'aplicació, que en aquest cas considerarem que és la meva directora de PFC, la Carme Quer, es van obtenir altres millores a desenvolupar a l'eina.

Procés 3

Aquest procés ha estat desenvolupat completament per mi i ha consistit en la implementació de totes les mancances i millores detectades al procés 2.

Procés 4

Aquest procés s'ha anat desenvolupant conjuntament amb la 3^a, ja que a l'aplicar-se una metodologia àgil, les funcionalitats eren especificades i acte seguit implementades per tal d'obtenir la validació corresponent. Un cop implementada cada millora especificada aquesta implementació es validava, i es prosseguia amb l'especificació i implementació de la funcionalitat següent.

Els processos 5 i 6 no han sigut necessaris en el manteniment de l'aplicació ja que el software encara no estava implantat, i per tant la seva migració o retirada no han sigut necessàries.

5.2. Tipus de manteniment de software

La tipologia del manteniment del software segons la seva finalitat està definida en quatre grans grups [WEB ISO 14764]:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

El desenvolupament de la nova versió comportarà el desenvolupament de tots els tipus de manteniment anteriors, excepte de l'Adaptive Maintenance, ja que en el cas de l'aplicació *RPToolManager* no és necessari.



Figura 7: Tipus de manteniment de software

5.2.1. Corrective Maintenance

Manteniment desenvolupat per a corregir errors en hardware o software. Es tracta bàsicament de solucionar tots els bugs no tractats en la versió 1 de l'aplicació. A tal efecte, es comproven totes les funcionalitats amb tots els casos possibles d'interacció per assegurar la fiabilitat del producte. Un cop identificats, són resolts amb la implementació pertinent.

5.2.2. Adaptive Maintenance

És el manteniment de software desenvolupat per fer que una aplicació sigui usable en un entorn que ha canviat. Aquesta part tracta sobre portar i migrar l'aplicació a una altra arquitectura. En el nostre cas, això no serà necessari.

5.2.3. Perfective Maintenance

És el manteniment de software desenvolupat per millorar les prestacions, la mantenibilitat i altres atributs d'una aplicació. En el nostre cas, consistirà en l'addició de noves funcionalitats i en la millora de les funcionalitats existents en *RPToolManager*.

5.2.4. Preventive Maintenance

Es tracta bàsicament de millorar la documentació de totes les funcionalitats, per tal de que si és necessari un manteniment sobre aquestes sigui fàcil pel programador saber quines són les operacions que s'han d'utilitzar. En aquest sentit, totes les operacions de l'aplicació estaran documentades, seguint l'estàndard que proporciona el Java a tal efecte, anomenat JavaDoc.

6. Anàlisi de la 1ª versió de RPToolManager

1.	Introducció.....	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1ª versió de RPToolManager	47
6.1.	Propòsit de l'anàlisi	48
6.2.	Desenvolupament de l'anàlisi	48
6.3.	Conclusions	49
6.3.1.	Errors a nivell de domini.....	49
6.3.2.	Errors a nivell d'interfície	50
6.3.3.	Mancances	57
6.3.4.	Noves funcionalitats	59
7.	Model Conceptual	61
8.	Base de Dades.....	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

6.1. Propòsit de l'anàlisi

L'anàlisi de l'aplicació *RPToolManager* pretén detectar tots aquells aspectes que són necessaris modificar en l'aplicació per tal d'aconseguir una 2^a versió lliure d'errors i mancances, i amb noves funcionalitats que poden facilitar l'ús d'aquesta. L'anàlisi, per tant, pretén trobar tot allò que falta a la 1^a versió de l'aplicació per tal de disposar d'una nova versió estable i llesta per ser utilitzada sense cap problema.

L'anàlisi intentarà identificar quatre tipus d'aspectes:

- Errors detectats a nivell de domini.
- Errors detectats a nivell d'interfície.
- Mancances actuals que l'aplicació té.
- Noves funcionalitats.

6.2. Desenvolupament de l'anàlisi

Aquest apartat fa referència molt breument a com s'ha desenvolupat l'anàlisi de l'aplicació *RPToolManager*.

Per tal de trobar els errors a nivell de domini i d'interfície, partint del document realitzat per l'Oriol Collell en un anàlisi previ, el que es va fer és detectar tots els bugs de l'aplicació provant per cada funcionalitat els diferents comportaments que produïa l'aplicació. Més concretament, es va:

- Per cada atribut de cada formulari, provar tots els seus possibles valors: valors buits o espais en blanc o massa llargs per aquest atribut, amb caràcters no comuns ('ñ', 'ç', signes de puntuació, '<' i '>') o incorrectes (especialment quan l'atribut havia de tenir format numèric) i el comportament amb valors ja introduïts en el mateix camp d'un altre objecte del mateix tipus (noms repetits, etc.), si era rellevant pel camp.
- Per tots els valors testejats de cada atribut, comprovar que la resta de l'aplicació s'actualitzés correctament (arbre de navegació en la finestra d'edició de patrons, les finestres taronges i els arbres de les finestres de patrons i mètriques), així com també la Base de Dades.
- Per cada botó de cada finestra i per cada element dels menús emergents, comprovar que en tots els casos possibles, el comportament sempre fos el desitjat.

- Per últim, comprovar que la gestió tant de patrons com de mètriques com de sources i keywords funcionés correctament (que les altes, baixes i modificacions es produïen correctament i que la Base de Dades no quedava inconsistent).

Les diferents mancances i noves funcionalitats de l'aplicació es van obtenir en dues etapes:

- Inicialment, mitjançant la meua pròpia experiència amb l'aplicació. Tal i com es va anar desenvolupant la fase anterior d'anàlisi de l'aplicació, degut a la gran quantitat de proves que vaig haver de fer, es va obtenir una llista inicial de les mancances i noves funcionalitats que serien útils en l'aplicació.
- En una etapa posterior, a través d'entrevistes amb la client de l'aplicació, Carme Quer, aquesta llista inicial va ser extesa, obtenint així la llista definitiva de noves funcionalitats i mancances.

6.3. Conclusions

A continuació es llisten tots els aspectes que s'han obtingut després de l'anàlisi que s'han de modificar/afegir a la nova versió. Aquests aspectes es mostren dividits en els quatre grups o tipus indicats anteriorment.

6.3.1. Errors a nivell de domini

Aquests han sigut els errors o situacions detectats on l'aplicació no es comporta de la manera adequada, i que fan referència a la part de domini de l'aplicació:

- D1. En un requirement form, es permet afegir més d'una extension amb el mateix nom.
- D2. Dins d'un classification schema, es pot crear més d'un root classifier amb el mateix nom.
- D3. Dins d'un root classifier o d'un decomposed classifier, es pot crear més d'un basic/decomposed classifier amb el mateix nom.
- D4. Quan estàs creant/modificant una mètrica de tipus string, deixa afegir com a default value un string buit o un string que sigui tot espais en blanc.
- D5. Quan estàs creant/modificant una mètrica de tipus float, deixa afegir valors al mínim i al màxim que no compleixen la restricció de que el mínim ha de ser menor o igual que el màxim.

- D6. Quan estàs creant/modificant una mètrica de tipus float, deixa afegir valors al default value que no compleixen la restricció de que aquest valor ha d'estar comprès entre el mínim i el màxim.
- D7. Quan estàs creant/modificant una mètrica de tipus integer, deixa afegir valors al mínim i al màxim que no compleixen la restricció de que el mínim ha de ser menor o igual que el màxim.
- D8. Quan estàs creant/modificant una mètrica de tipus integer, deixa afegir valors al default value que no compleixen la restricció de que aquest valor ha d'estar comprès entre el mínim i el màxim.

6.3.2. Errors a nivell d'interfície

Aquests han sigut els errors detectats que fan referència a la part d'interfície de l'aplicació:

- I1. A la finestra de patrons i de mètriques, no funciona el botó per minimitzar la finestra.
- I2. A la finestra de patrons i de mètriques, quan el nom del patró o de la mètrica, respectivament, és molt llarg, no es veu sencer perquè no apareix l'scrollbar horitzontal; s'ha de modificar la mida de la finestra manualment per tal de veure el nom sencer.
- I3. A la finestra de patrons, al desplegable per seleccionar l'esquema no es veuen sencers els noms llargs.
- I4. A la finestra de patrons, en l'esquema *Alphabetical list* els patrons no apareixen ordenats alfabèticament.
- I5. A la finestra de patrons, a l'esborrar un classification schema ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del classification schema esborrat.
- I6. A la finestra de patrons, al modificar un root classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació d'abans de la modificació.
- I7. A la finestra de patrons, a l'esborrar un root classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del root classifier esborrat.
- I8. A la finestra de patrons, al modificar un basic/decomposed classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació d'abans de la modificació.

- 19. A la finestra de patrons, a l'esborrar un basic/decomposed classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del basic/decomposed classifier esborrat.
- I10. A la finestra de patrons, a l'esborrar un patró la informació del mateix persisteix a la part inferior taronja de la finestra tot i que el patró ja ha sigut esborrat.
- I11. A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb un nom que és un string buit o un conjunt d'espais en blanc, no deixa guardar però tampoc es mostra cap missatge d'error.
- I12. A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I13. A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb el nom d'un classification schema que ja existeix, no deixa guardar però tampoc es mostra cap missatge d'error.
- I14. A la finestra d'edició de classifiers, quan es vol guardar un classifier (ja sigui root, decomposed o basic) amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I15. A la finestra d'edició de root classifiers, la finestra té com a títol "classifier" en comptes de "root classifier".
- I16. A la finestra d'edició de patrons, quan es vol guardar un patró amb un nom que és un string buit o una cadena d'espais en blanc, la finestra que es mostra amb l'error no és gaire aclaridora.
- I17. A la finestra d'edició de patrons, quan es vol guardar un patró amb el nom d'un altre patró que ja existeix, la finestra que es mostra amb l'error no és gaire aclaridora.
- I18. A la finestra d'edició de patrons, quan es vol guardar un patró amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I19. A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp *author* és massa llarg, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I20. A la finestra d'edició de patrons, a la pestanya attributes del patró, al calendari del camp *version*, a la capsa de text per introduir l'any, si introduïm anys més grans de 9999, tot i ser anys no vàlids per l'aplicació, surten escrits en color verd volent dir que són vàlids.

- I21. A la finestra d'edició de patrons, a la pestanya dependencies, es pot afegir la mateixa dependency més d'un cop, tot i que després no es deixa guardar els canvis.
- I22. A la finestra d'edició de patrons, quan es vol guardar un patró que té dues dependències iguals, el missatge d'error que apareix és incorrecte (surt el mateix error que quan el nom del patró ja existeix).
- I23. A la finestra per seleccionar sources (a la qual s'accedeix des de la finestra d'edició de patrons o de mètriques quan es vol afegir una nova source), el missatge d'error que es mostra quan s'intenta esborrar una source que està associada a un patró o a una mètrica és incorrecte.
- I24. A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), no es mostra cap error quan l'identificador de la source és massa llarg com per guardar-lo a la Base de Dades.
- I25. A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), no es mostra cap error quan l'identificador de la source ja existeix.
- I26. A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), si vols cancel·lar abans d'haver introduït un identificador per la source no et deixa, et demana que introdueixis un identificador.
- I27. A la finestra per seleccionar keywords (a la qual s'accedeix des de la finestra d'edició de patrons quan es vol afegir una nova keyword), el missatge d'error que es mostra quan s'intenta esborrar una keyword que està associada a un patró és incorrecte.
- I28. A la finestra per editar keywords (a la qual s'accedeix des de la finestra per seleccionar keywords), es deixa introduir noves keywords que siguin només espais en blancs o modificar una keyword a només espais en blancs.
- I29. A la finestra per editar keywords (a la qual s'accedeix des de la finestra per seleccionar keywords), el missatge que surt quan una keyword és massa llarga per guardar-la a la Base de Dades o quan la keyword ja existeix és el mateix; estaria bé diferenciar-los.
- I30. A la finestra d'edició de patrons, quan en el formulari que surt quan es prem el botó *New Form...* s'introdueix un string buit o un seguit d'espais en blanc, l'aplicació no crea el nou form però tampoc mostra cap error.
- I31. A la finestra d'edició de patrons, es pot modificar el nom d'un form pel d'un altre form que ja existeix en el patró o canviar-lo a un string buit i es pot guardar sense cap problema, i això no hauria de ser possible.
- I32. A la finestra d'edició de patrons, quan es vol guardar un patró que té un requirement form amb un nom massa llarg, i per tant no deixa guardar-lo a la

Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.

- I33. A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp *author* d'un dels seus requirement forms és massa llarg, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I34. A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp *form text* d'una de les seves fixed/extended part d'un dels seus requirement forms és buit, la finestra que es mostra amb l'error no és gaire aclaridor.
- I35. A la finestra d'edició de patrons, quan es crea un nou requirement form i al form text de la fixed part es posen només espais en blancs o si quan ja està creat es modifica aquest form text a un string buit o a un conjunt d'espais, deixa guardar el patró i això és incorrecte.
- I36. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan es crea una nova extended part dins d'un form i al seu form text es posen només espais en blancs o si quan ja està creat es modifica aquest form text a un string buit o a un conjunt d'espais, deixa guardar el patró i això és incorrecte.
- I37. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan el nom d'una extension és més gran del que cap al desplegable per seleccionar l'extended part, el nom no apareix sencer i el desplegable acaba ocupant tota la finestra, fent desaparèixer la part de navigation de la finestra d'edició de patrons.
- I38. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan s'intenta afegir una nova extended part amb un nom buit, no s'afegeix l'extended part però no es mostra cap error com que no s'ha afegit.
- I39. A la finestra d'edició de patrons, quan es vol guardar un patró que té una extended part amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I40. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan s'esborra una extended part, si aquesta era l'última que quedava, es queden les seves dades als camps del formulari, tot i que l'extension s'ha eliminat.
- I41. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts i es vol afegir un nou paràmetre, el sistema deixa que introduïm un paràmetre sense nom.
- I42. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts i es vol afegir un nou paràmetre, el

missatge d'error que surt quan el nom del paràmetre ja existeix en el fixed o extended part no és gaire aclaridor.

- I43. A la finestra d'edició de patrons, l'aplicació permet que quan estàs modificant la informació d'un paràmetre, es canviï el nom del paràmetre al d'un altre paràmetre que ja existeix dins de la mateixa fixed/extended part, o que es canviï el nom a un string buit o a un conjunt d'espais en blanc.
- I44. A la finestra d'edició de patrons, quan es vol guardar un patró que té un paràmetre amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.
- I45. A la finestra d'edició de patrons, quan es vol guardar un patró que té un paràmetre sense mètrica assignada, el missatge d'error que es mostra no és gaire aclaridor.
- I46. Cada cop que li donem al botó per obrir i tancar la finestra de mètriques, aquesta surt en una posició diferent.
- I47. A la finestra de mètriques, quan deixem el cursor a sobre d'un element de l'arbre surt al quadradet groc típic però té escrit un text incorrecte.
- I48. A la finestra de mètriques, a la informació sobre la mètrica que hi ha seleccionada que surt a la part inferior taronja de la finestra, si el nom de la mètrica conté el caràcter '<', només surt el nom fins al caràcter '<'. Ídem pel caràcter '>'.
- I49. A la finestra de mètriques, si fas clic dret a sobre d'una mètrica de tipus time point no funciona l'opció del menú emergent *Modify...*, no fa res.
- I50. A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que no hi ha nom a la mètrica, però tampoc et deixa guardar.
- I51. A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que el nom de la mètrica és massa llarg, però tampoc et deixa guardar.
- I52. A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que ja existeix una mètrica amb el mateix nom que aquesta, però tampoc et deixa guardar.
- I53. A la finestra d'edició de mètriques de tipus integer, si estàs modificant una mètrica, l'aplicació no et deixa guardar els canvis tot i que siguin valors correctes. Per exemple, si es vol modificar un *integer* que s'ha creat amb mínim -12 i màxim -11 i es vol modificar i posar mínim 12 i màxim 16, tota l'estona dóna error (si es crea una nova mètrica amb aquests mínims i màxims si que la crea bé).

- I54. A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que no hi ha cap mínim assignat o de que aquest no té format numèric d'enter, però tampoc et deixa guardar.
- I55. A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que no hi ha cap màxim assignat o de que aquest no té format numèric d'enter, però tampoc et deixa guardar.
- I56. A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que quan estant seleccionada la casella per introduir el default value no hi ha cap default value assignat o el default value introduït no té format numèric d'enter, però tampoc et deixa guardar.
- I57. A la finestra d'edició de mètriques de tipus float, si estàs modificant una mètrica, l'aplicació no et deixa guardar els canvis tot i que siguin valors correctes. Per exemple, si es vol modificar un *float* que s'ha creat amb mínim -12 i màxim -11 i es vol modificar i posar mínim 12 i màxim 16, tota l'estona dona error (si es crea una nova mètrica amb aquests mínims i màxims si que la crea bé).
- I58. A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que no hi ha cap mínim assignat o de que aquest no té format numèric de float, però tampoc et deixa guardar.
- I59. A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que no hi ha cap màxim assignat o de que aquest no té format numèric de float, però tampoc et deixa guardar.
- I60. A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que quan estant seleccionada la casella per introduir el default value no hi ha cap default value assignat o el default value introduït no té format numèric de float, però tampoc et deixa guardar.
- I61. A la finestra d'edició de mètriques de tipus string, no et mostra cap error quan es viola la condició de longitud màxima del default value, però tampoc et deixa guardar.
- I62. A la finestra d'edició de mètriques de tipus time point, el formulari té com a títol "new string metric".
- I63. A la finestra d'edició de mètriques de tipus time point, el camp *day* del formulari deixa introduir com a màxim fins el dia 31, cosa que està bé, però no tots els mesos tenen 31 dies. S'hauria de comprovar això quan s'introdueix també un mes.
- I64. A la finestra d'edició de mètriques de tipus time point, si s'introdueix al camp *hour* o *minute* un nombre superior al màxim permès, en lloc de substituir-se pel nombre més gran (23 o 59), es substitueix per un nombre que no té ni cap ni peus que sigui aquell nombre. Passa el mateix amb el camp *day* i *second*.

- I65. A la finestra d'edició de mètriques de tipus domain, quan s'està modificant una mètrica, no deixa guardar els canvis mai, encara que tots els valors introduïts siguin correctes.
- I66. A la finestra d'edició de mètriques de tipus domain, deixa guardar la mètrica tot i que no hi hagi cap value afegit, i això no hauria de ser possible.
- I67. A la finestra d'edició de mètriques de tipus domain, deixa afegir valors buits, i això no hauria de ser possible.
- I68. A la finestra d'edició de mètriques de tipus domain, deixa afegir valors molt llargs, i això no hauria de ser possible, ja que després quan vols guardar la mètrica no et deixa guardar-la però tampoc mostra l'error que es produeix.
- I69. A la finestra d'edició de mètriques de tipus domain, es deixa afegir dos o més valors possibles iguals, però després no et deixa guardar i l'error que et mostra només diu que hi ha algun camp incorrecte.
- I70. A la finestra d'edició de mètriques de tipus set, no et mostra l'error quan no hi ha cap mètrica simple assignada, però tampoc et deixa guardar.
- I71. No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:
1. Estem editant un patró.
 2. Afegim una nova dependency amb qualsevol patró.
 3. Esborrem el patró de la dependency afegida.
 4. Intentem guardar.
- I72. No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:
1. Estem editant un patró.
 2. Afegim un nou paràmetre a qualsevol fixed o extended part d'un dels seus requirement forms i li assignem una mètrica qualsevol.
 3. Esborrem la mètrica que hem assignat al paràmetre.
 4. Intentem guardar.
- I73. No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:
1. Estem modificant un patró.
 2. L'esborrem.
 3. Intentem guardar.
- I74. No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:
1. Estem editant una mètrica de tipus set.
 2. Assignem qualsevol mètrica simple com a tipus del set.
 3. Esborrem la mètrica que hem assignat al tipus del set.
 4. Intentem guardar.

6.3.3. Mancances

Aquestes han sigut les mancances detectades durant l'anàlisi de l'aplicació:

- M1. Faria falta un botó similar al que obre la finestra de mètriques per obrir la finestra de patrons, perquè si es tanca aquesta finestra no hi ha manera de tornar-la a obrir sense tancar l'aplicació i tornar-la a executar.
- M2. A la finestra de patrons, al crear un nou classification schema, el desplegable del classification schema seleccionat no canvia al nou ítem creat.
- M3. A la finestra de patrons, al crear un nou classification schema, el desplegable que conté tots els classifications schema es desordena. A més, l'ordenació del desplegable per defecte no es fa diferenciant entre majúscules i minúscules.
- M4. A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat hauria de sortir també el goal, ja que a aquest camp se li ha donat molta importància en la documentació del projecte, però poca a l'aplicació.
- M5. A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat, les sources queden ordenades de forma aleatòria. Haurien d'aparèixer ordenades alfabèticament.
- M6. A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat, l'ordenació dels requirement forms d'aquest és aleatòria, ja que conforme vas modificant la informació dels forms es canvia l'ordenació, i tampoc es correspon amb l'ordenació que hi ha al formulari d'edició del patró.
- M7. A la finestra d'edició de patrons, a l'arbre de navegació del patró, no es veuen ni els forms ni les seves extensions ordenades alfabèticament. Per facilitar la navegació per l'arbre, hauria de ser així.
- M8. A la finestra d'edició de patrons, a la pestanya attributes, les keywords no queden ordenades de cap manera i de fet cada cop que es canvia de pestanya l'ordenació canvia, i això és un embolic. S'haurien d'ordenar alfabèticament.
- M9. A la finestra d'edició de patrons, a la pestanya forms, els requirement forms no queden ordenats de cap manera. S'haurien d'ordenar alfabèticament.
- M10. A la finestra per seleccionar keywords (a la qual s'accedeix des de la finestra d'edició de patrons quan es vol afegir una nova keyword), les keywords no queden ordenades de cap manera i això dificulta la cerca d'una en concret. S'haurien d'ordenar alfabèticament.
- M11. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts, s'hauria d'explicar breument en

una banda com funciona el botó *Extract from text* (dient que els paràmetres s'han de posar entre %...%).

- M12. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya *extended parts*, al desplegable per seleccionar l'extension que volem editar aquestes surten ordenades sense cap mena de criteri, i això dificulta la cerca quan hi ha unes quantes extensions. Haurien d'aparèixer ordenades alfabèticament.
- M13. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya *extended parts*, quan es crea una nova extension no queda seleccionada al desplegable.
- M14. A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya *fixed* o *extended parts*, hauria d'haver un acord en el format en el que poden estar els paràmetres d'una *fixed* o *extended part* d'un requirement form, ja que:
- 1) Si fem *New parameter...* deixa introduir el que vulguem com a nom de paràmetre.
 - 2) Si ho fem pel mètode *Extract from text* no fa ni cas dels paràmetres que tinguin algun caràcter que no sigui una lletra.
- S'hauria d'unificar l'edició dels noms dels paràmetres en ambdues opcions.
- M15. A la finestra de mètriques, a la informació que surt a la part inferior taronja de la finestra sobre la mètrica que hi ha seleccionada, les sources queden ordenades de forma aleatòria. Haurien d'aparèixer ordenades alfabèticament.
- M16. A la finestra de mètriques, igual que a la finestra de patrons hi ha dos botons que es diuen *Expand all* i *Collapse all*, en aquesta finestra també hi podrien ser.
- M17. Al formulari d'edició de mètriques de tipus float, seria útil indicar quin símbol s'utilitza pels decimals.
- M18. Al formulari d'edició de mètriques de tipus time point, quan estàs creant una nova mètrica, hauria de sortir la data/hora actual.
- M19. Al formulari d'edició de mètriques de tipus time point, el mes es selecciona per nom. En canvi, al camp taronja de la finestra de mètriques, quan selecciones una mètrica de tipus time point, el mes surt com a número. S'hauria d'unificar.
- M20. Al formulari d'edició de mètriques de tipus time point, el mes es selecciona per nom en castellà. En canvi, tota la resta de l'aplicació és en anglès. S'hauria d'unificar.
- M21. Al formulari d'edició de mètriques de tipus time point, no surt l'any. De fet, una mètrica de tipus time point no té l'atribut any. S'hauria d'afegir.

- M22. Al formulari d'edició de mètriques de tipus time point, seria útil utilitzar un servei de calendari (per les dates) i un desplegable o rellotge per les hores.
- M23. Al formulari d'edició de mètriques de tipus set, als camps *Selected metric* i *Current metric* es pot seleccionar el valor que tenen, però de fet no és útil aquesta selecció i potser pot portar a confusió. Seria millor si fossin camps bloquejats.
- M24. Al formulari d'edició de mètriques de tipus set, quan es prem el botó *Assign* no fa res. S'hauria d'avisar de que s'ha de seleccionar una mètrica simple a la finestra de mètriques.

6.3.4. Noves funcionalitats

A continuació es llisten les noves funcionalitats que poden ser útils de cara a facilitar la utilització de l'eina, trobades mentre es realitzava l'anàlisi de l'aplicació:

- F1. Seria interessant poder veure tota la informació del patró en una mateixa finestra (similar a com es mostra a la Taula 1 d'aquest document, a l'apartat 2.3 Estructura d'un patró de requisit).
- F2. Seria interessant poder veure tota la informació d'una mètrica en una mateixa finestra.
- F3. Seria útil poder fer una copia d'un classification schema sencer.
- F4. En la finestra d'edició de patrons, seria útil tenir una nova pestanya quan el patró està seleccionat en l'arbre de navegació de la finestra, amb l'objectiu de poder gestionar a quins classifilers pertany un patró dins d'un classification schema.

7. Model Conceptual

1.	Introducció	11
2.	Patrons de requisits	15
3.	Eines de Treball	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
7.1.	<i>Model conceptual de la 1^a versió</i>	<i>62</i>
7.1.1.	Diagrama de classes	62
7.1.2.	Restriccions d'integritat textuals	65
7.2.	<i>Nou model conceptual</i>	<i>66</i>
8.	Base de Dades	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1^a versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades	195
18.	Document de Disseny	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

7.1. Model conceptual de la 1^a versió

En aquest apartat, s'explica com va ser ideat i dissenyat el model conceptual de la versió 1 de l'eina *RPToolManager*, per així poder descriure després les modificacions que van ser necessàries per la 2^a versió de l'aplicació.

7.1.1. Diagrama de classes

A les Figures 8 i 9 es mostra el diagrama de classes de la 1^a versió de l'aplicació *RPToolManager*; per a que sigui més fàcil de seguir, s'ha separat en dues parts: la de patrons i la de mètriques. En aquest diagrama podem trobar les diferents classes d'objectes, que podem descriure agrupades en les categories següents:

- *Source*. Representa les fonts d'informació d'on es poden obtenir els diferents objectes que formen part del model.
- *Classification Schema*. Representa els diferents esquemes de classificació que pot tenir un catàleg de patrons de requisits. Un classification schema està format per varis *Root Classifiers*; un root classifier, però, no és més que un subtipus de la classe *Decomposed Classifier*. Per un altre banda, tenim la classe *General Classifier*, que representa a tots els tipus de classificadors que pot tenir un root classifier, que són: o bé *Basic Classifiers* (que només poden contenir patrons) o bé *Decomposed Classifiers* (que només poden contenir altres basic o decomposed classifiers). L'atribut *pos* de la classe *General Classifier* representa la posició d'aquest dins del seu classificador pare.
- *Keyword*. Representa les paraules clau que poden caracteritzar un patró.
- *Requirement Pattern*. Representa a un patró de requisit. Un patró pot tenir dependències amb altres patrons i té un o més *Requirement Form*, que al seu cop està format per una *Fixed Part* i zero o més *Extended Part*. Aquestes dues classes hereten de *Pattern Item*, ja que els objectes d'ambdues poden tenir un question text, un form text i zero o més *Parameter*. Un patró pot tenir varies *Keyword*.
- *Parameter*. Representa un paràmetre que pertany a una fixed o extended part. A més del nom i la descripció, té l'atribut *correctnessCondition* que s'utilitza per limitar els valors correctes que el paràmetre pot prendre. A més, un paràmetre sempre té associada una *Metric*.
- *Metric*. Representa les mètriques que limiten quins són els valors correctes per un paràmetre. Una mètrica pot ser de tipus *Simple* (un únic valor) o *Set* (un conjunt de valors d'una mètrica simple). Les mètriques simples poden ser o de tipus *Integer* o *Float* o *String* o *Time Point* (un instant) o *Domain* (un valor dins d'un conjunt de valors). Totes les mètriques simples, excepte la time point, poden tenir o no valor per defecte.

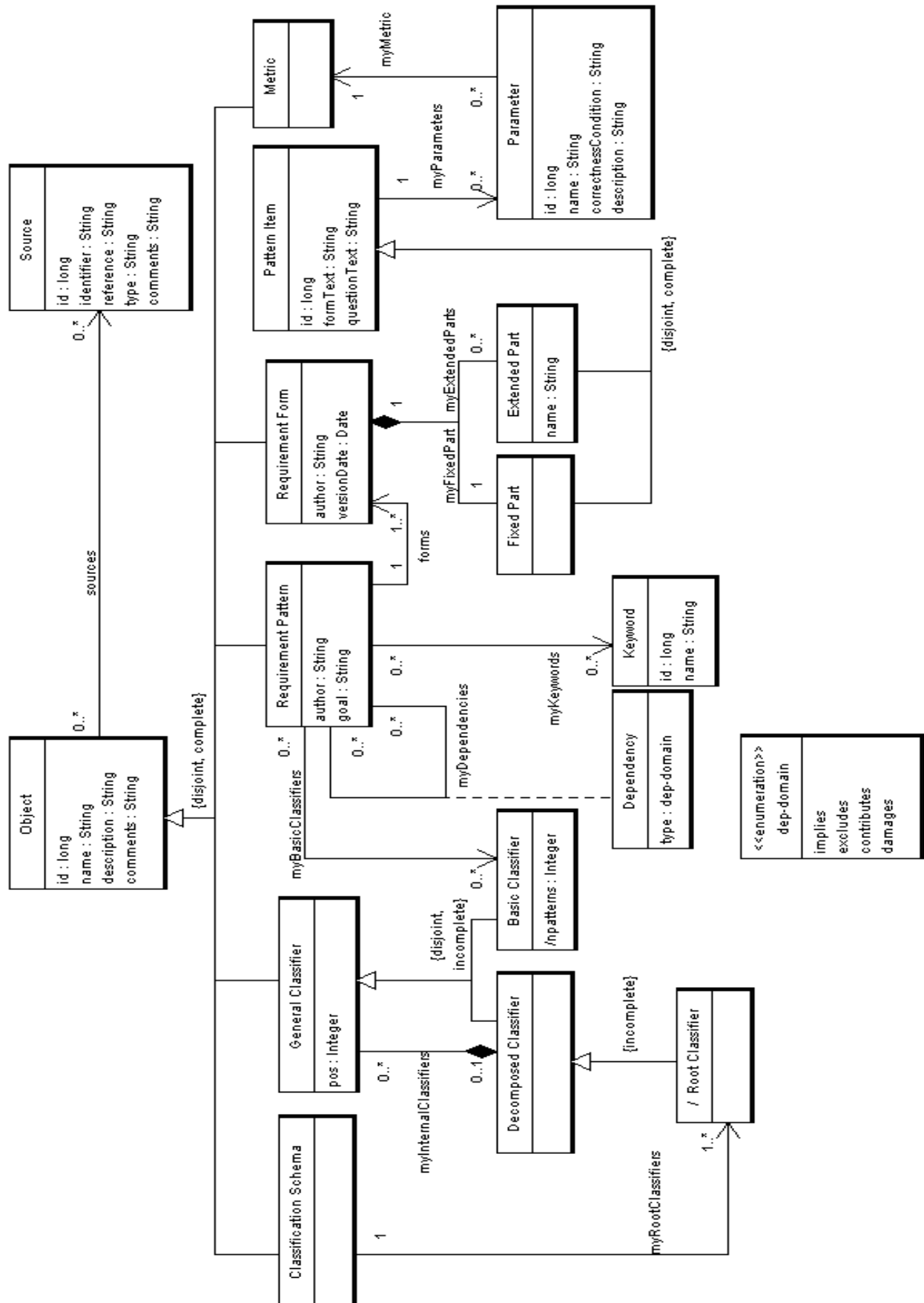


Figura 8: Diagrama de classes, part de patrons

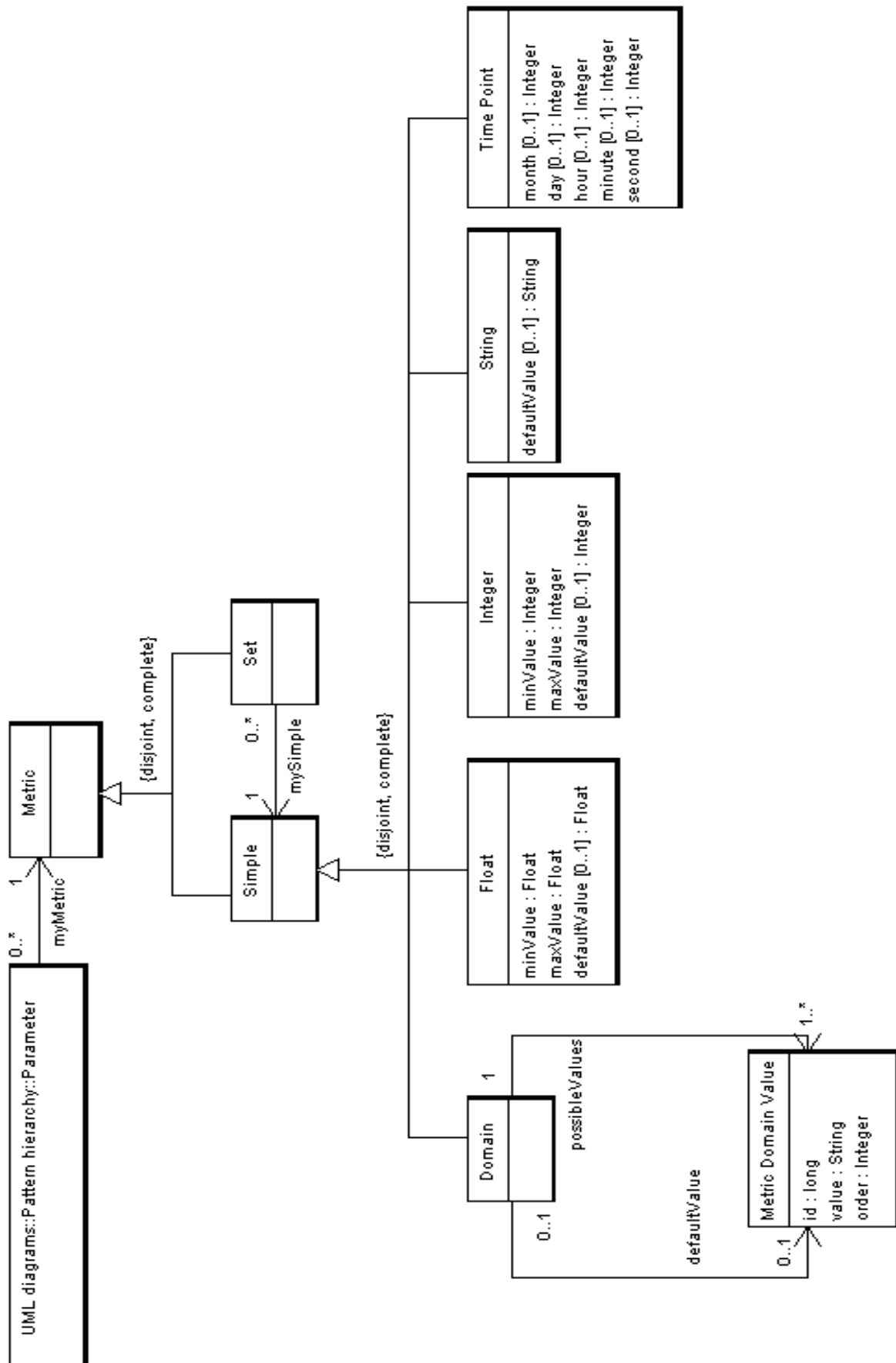


Figura 9: Diagrama de classes, part de mètriques

7.1.2. Restriccions d'integritat textuais

Aquestes són les restriccions d'integritat textuais necessàries per una descripció precisa del domini i que no han pogut ser expressades gràficament en els diagrames anteriors:

- No pot haver-hi dues *Source* amb el mateix *id*.
- No pot haver-hi dues *Source* amb el mateix *identifier*.
- No pot haver-hi dos *Generic Object* amb el mateix *id*.
- No pot haver-hi dos *Classification Schema* amb el mateix *name*.
- Dins del mateix *Classification Schema*, no pot haver-hi dos *Root Classifier* amb el mateix *name*.
- Dins del mateix *Decomposed Classifier*, no pot haver-hi dos *General Classifier* amb el mateix *name*.
- L'atribut derivat *npatterns* d'un *Basic Classifier* és igual al nombre de *Requirement Pattern* que té associats.
- No pot haver-hi dos *Keyword* amb el mateix *id*.
- No pot haver-hi dos *Keyword* amb el mateix *name*.
- No pot haver-hi dos *Requirement Pattern* amb el mateix *name*.
- Dins del mateix *Requirement Pattern*, no pot haver-hi dos *Requirement Form* amb el mateix *name*.
- Dins del mateix *Requirement Form*, no pot haver-hi dos *Extended Part* amb el mateix *name*.
- No pot haver-hi dos *Pattern Item* amb el mateix *id*.
- Dins del mateix *Pattern Item*, no pot haver-hi dos *Parameter* amb el mateix *name*.
- No pot haver-hi dos *Metric* amb el mateix *name*.
- Una mètrica tipus *Integer* ha de complir en els seus atributs que $minValue \leq maxValue$ i, en cas que tingui un valor per defecte definit, $minValue \leq defaultValue \leq maxValue$.
- Una mètrica tipus *Float* ha de complir en els seus atributs que $minValue \leq maxValue$ i, en cas que tingui un valor per defecte definit, $minValue \leq defaultValue \leq maxValue$.

- Una mètrica de tipus *Domain* ha de complir que, si té un *defaultValue* associat, aquest ha de ser un dels seus *possibleValues*.
- L'atribut *name* d'un *Generic Object*, *Extended Part*, *Parameter* o *Keyword* i l'atribut *identifier* d'una *Source* no pot ser un string buit ni un conjunt d'espais en blanc.
- L'atribut *formText* d'un *Pattern Item* no pot ser un string buit ni un conjunt d'espais en blanc.

7.2. Nou model conceptual

Pel que fa al model conceptual de la 2^a versió de l'aplicació *RPToolManager*, l'única diferència que hi ha respecte al de la 1^a versió és que s'ha afegit un nou atribut a la subclasse *Time Point* d'una mètrica simple per tal que també es pugui definir l'any. Així doncs, la nova classe *Time Point* queda tal com segueix al nou model conceptual:

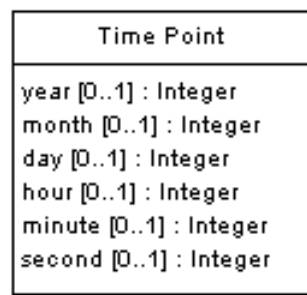


Figura 10: Diagrama de classes, nova classe Time Point

La resta del model conceptual queda exactament igual.

8. Base de Dades

1.	Introducció	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades.....	67
8.1.	<i>Introducció</i>	<i>68</i>
8.2.	<i>Modificació del mapeig per Hibernate.....</i>	<i>68</i>
8.3.	<i>Modificació de la Base de Dades existent.....</i>	<i>73</i>
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1^a versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work.....	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

8.1. Introducció

En aquest apartat explicarem quines són les modificacions que van ser necessàries a la Base de Dades que hi ha darrera de l'aplicació degut al canvi realitzat al model conceptual (per més informació sobre aquest canvi, veure l'apartat anterior 7.2).

8.2. Modificació del mapeig per Hibernate

Per tal de fer funcionar Hibernate, entre d'altres, fa falta un fitxer .hbm, més conegut com fitxer de mapeig, que permet al programador definir com és el seu model de dades, quines relacions existeixen i quina forma tenen.

Així doncs, degut al nou atribut *year* de la subclasse *Time Point* de la classe mètrica, va ser necessari modificar el fitxer de mapeig per afegir aquest nou atribut.

A continuació es mostra com és el contingut del fitxer de mapeig per la 2^a versió de l'aplicació *RPToolManager* (el de la 1^a versió era exactament igual, eliminant la línia verd).

```
<hibernate-mapping>

  <class name="edu.upc.gessi.rptool.domain.requirementpatterns.Keyword"
    table="KEYWORD">
    <id name="id" column="ID" type="long">
      <generator class="native"/>
    </id>
    <property name="name" column="NAME" unique="true" not-null="true"/>
  </class>

  <class name="edu.upc.gessi.rptool.domain.Source" table="SOURCE">
    <id name="id" column="ID" type="long">
      <generator class="native"/>
    </id>
    <property name="identifier" column="IDENTIFIER" unique="true" not-
      null="true"/>
    <property name="reference" type="text" length="2000" column="REFERENCE"
      not-null="true"/>
    <property name="type" type="text" length="2000" column="TYPE" not-
      null="true"/>
    <property name="comments" type="text" length="2000" column="COMMENTS"
      not-null="true"/>
  </class>

  <class name="edu.upc.gessi.rptool.domain.requirementpatterns.Parameter"
    lazy="false" table="PARAMETER">
    <id name="id" column="ID" type="long">
      <generator class="native"/>
    </id>
    <property name="name" column="NAME" unique="false" not-null="true"/>
    <property name="correctnessCondition" column="CORRECTNESS_CONDITION"
      not-null="true"/>
  </class>
```

```

    <property name="description" column="DESCRIPTION" not-null="true"/>
    <many-to-one name="myMetric" column="MY_METRIC" not-null="true"/>
</class>

<class name="edu.upc.gessi.rptool.domain.requirementpatterns.PatternItem"
lazy="false" table="PATTERN_ITEM">

    <id name="id" column="ID" type="long">
        <generator class="native"/>
    </id>

    <property name="patternText" column="PATTERN_TEXT" not-null="true"/>
    <property name="questionText" column="QUESTION_TEXT" not-null="false"/>

    <set name="myParameters" lazy="false">
        <key column="MY_PATTERN_ITEM"/>
        <one-to-many class="edu.upc.gessi.rptool.domain.requirementpatterns.
        Parameter"/>
    </set>

    <joined-subclass name="edu.upc.gessi.rptool.domain.requirementpatterns.
ExtendedPattern" lazy="false" table="EXTENDED_PATTERN">
        <key column="ID"/>
        <property name="name" column="NAME" not-null="true"/>
    </joined-subclass>

    <joined-subclass name="edu.upc.gessi.rptool.domain.requirementpatterns.
FixedPattern" lazy="false" table="FIXED_PATTERN">
        <key column="ID"/>
    </joined-subclass>

</class>

<class name="edu.upc.gessi.rptool.domain.metrics.MetricDomainValue"
lazy="false" table="METRICS_DOMAIN_VALUE">
    <id name="id" column="ID" type="long">
        <generator class="native"/>
    </id>
    <property name="value" column="VALUE" not-null="true"/>
    <property name="order" column="VALUE_ORDER" not-null="true"/>
</class>

<class name="edu.upc.gessi.rptool.domain.requirementpatterns.
Dependency" table="DEPENDENCY">
    <id name="id" column="ID" type="long">
        <generator class="native"/>
    </id>
    <properties name="unique-constraint" unique="true">
        <property name="idType" type="integer" column="ID_TYPE" not-null="
        true"/>
        <property name="idFirst" type="long" column="ID_FIRST" not-null=
        "true"/>
        <property name="idSecond" type="long" column="ID_SECOND" not-
        null="true"/>
    </properties>
    <many-to-one name="first" column="FIRST_1" not-null="true"/>
    <many-to-one name="second" column="SECOND_2" not-null="true"/>
</class>

```

```

<class name="edu.upc.gessi.rptool.domain.GenericObject" table="GENERIC_
OBJECT">

  <id name="id" column="ID" type="long">
    <generator class="native"/>
  </id>
  <property name="name" column="NAME" not-null="true"/>
  <property name="description" type="text" length="2000" column=
"DESCRIPTION" not-null="true"/>
  <property name="comments" type="text" length="2000" column="COMMENTS"
not-null="true"/>

  <set name="sources" table="OBJECT_SOURCE" lazy="false">
    <key column="OBJECT_ID"/>
    <many-to-many column="SOURCE_ID" class="edu.upc.gessi.rptool.domain.
Source"/>
  </set>

  <joined-subclass name="edu.upc.gessi.rptool.domain.schema.
ClassificationSchema" table="CLASSIFICATION_SCHEMA">
    <key column="ID"/>
    <property name="name" column="NAME" not-null="true" unique="true"/>

    <set name="myRootClassifiers" lazy="false">
      <key column="CLASSIFICATION_CATALOGUE_ID"/>
      <one-to-many class="edu.upc.gessi.rptool.domain.schema.
RootClassifier"/>
    </set>

  </joined-subclass>

  <joined-subclass name="edu.upc.gessi.rptool.domain.schema.
InternalClassifier" table="GENERAL_CLASSIFIER">
    <key column="ID"/>
    <property name="npatterns" type="integer" column="NPATTERNS"/>
    <property name="pos" type="integer" column="POS"/>
    <property name="type" type="integer" column="TYPE" not-null="true"/>

    <set name="myInternalClassifiers" lazy="false">
      <key column="CLASSIFIER_ID"/>
      <one-to-many class="edu.upc.gessi.rptool.domain.schema.
InternalClassifier"/>
    </set>

    <joined-subclass name="edu.upc.gessi.rptool.domain.schema.
RootClassifier" table="ROOT_CLASSIFIER">
      <key column="ID"/>
    </joined-subclass>

  </joined-subclass>

  <joined-subclass name="edu.upc.gessi.rptool.domain.requirementpatterns.
RequirementForm" table="REQUIREMENT_FORM">
    <key column="ID"/>
    <property name="author" column="AUTHOR" not-null="true"/>
    <property name="version" type="timestamp" column="VERSION" not-
null="true"/>
    <many-to-one name="myFixedPattern" column="MY_FIXED_PATTERN"
unique="true" not-null="true"/>

    <set name="myExtendedPatterns" lazy="false">
      <key column="MY_EXTENDED_PATTERN"/>

```

```

        <one-to-many class="edu.upc.gessi.rptool.domain.
        requirementpatterns.ExtendedPattern"/>
    </set>

</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.requirementpatterns.
RequirementPattern" table="REQUIREMENT_PATTERN">
    <key column="ID"/>
    <property name="name" column="NAME" not-null="true" unique="true"/>
    <property name="author" column="AUTHOR" not-null="true"/>
    <property name="goal" type="text" length="2000" column="GOAL" not-
    null="true"/>

    <set name="forms" lazy="false">
        <key column="MY_REQUIREMENT_PATTERN"/>
        <one-to-many class="edu.upc.gessi.rptool.domain.
        requirementpatterns.RequirementForm"/>
    </set>

    <set name="dependencies" lazy="false">
        <key column="DEPENDENCIES"/>
        <one-to-many class="edu.upc.gessi.rptool.domain.
        requirementpatterns.Dependency"/>
    </set>

    <set name="myKeywords" table="REQ_PATTERN_KEYWORD" lazy="false">
        <key column="REQ_PATTERN_ID"/>
        <many-to-many column="KEYWORD_ID" class="edu.upc.gessi.rptool.
        domain.requirementpatterns.Keyword"/>
    </set>

    <set name="myBasicClassifiers" table="BASIC_PATTERN" lazy="false">
        <key column="REQ_PATTERN_ID"/>
        <many-to-many column="BASIC_ID" class="edu.upc.gessi.rptool.
        domain.schema.InternalClassifier"/>
    </set>

</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.Metric"
table="METRIC" lazy="false">
    <key column="ID"/>
    <property name="name" column="NAME" not-null="true" unique="true"/>

    <joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
SimpleMetric" table="SIMPLE_METRIC">
        <key column="ID"/>

    <joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
FloatMetric" table="FLOAT_METRIC">
        <key column="ID"/>
        <property name="minValue" type="float" column="MIN_VALUE" not-
        null="true"/>
        <property name="defaultValue" type="float" column="DEFAULT_
        VALUE" not-null="false"/>
        <property name="hasDefault" type="boolean" column="HAS_
        DEFAULT" not-null="false"/>
        <property name="maxValue" type="float" column="MAX_VALUE" not-
        null="true"/>
    </joined-subclass>

```

```

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
IntegerMetric" table="INTEGER_METRIC">
  <key column="ID"/>
  <property name="minValue" type="float" column="MIN_VALUE" not-
    null="true"/>
  <property name="defaultValue" type="float" column="DEFAULT_
    VALUE" not-null="false"/>
  <property name="hasDefault" type="boolean" column="HAS_
    DEFAULT" not-null="false"/>
  <property name="maxValue" type="float" column="MAX_VALUE" not-
    null="true"/>
</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
StringMetric" table="STRING_METRIC">
  <key column="ID"/>
  <property name="defaultValue" type="text" length="2000" column=
    "DEFAULT_VALUE" not-null="false"/>
</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
TimePointMetric" table="TIME_POINT_METRIC">
  <key column="ID"/>
  <property name="year" type="integer" column="YEAR_1" not-
    null="false"/>
  <property name="month" type="integer" column="MONTH" not-
    null="false"/>
  <property name="day" type="integer" column="DAY" not-
    null="false"/>
  <property name="hour" type="integer" column="HOURL_1" not-
    null="false"/>
  <property name="minute" type="integer" column="MINUTE_1" not-
    null="false"/>
  <property name="second" type="integer" column="SECOND_1" not-
    null="false"/>
</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
MetricDomain" lazy="false" table="METRIC_DOMAIN">
  <key column="ID"/>
  <set name="possibleValues" lazy="false">
    <key column="POSSIBLE_VALUE"/>
    <one-to-many class="edu.upc.gessi.rptool.domain.metrics.
      MetricDomainValue"/>
  </set>
  <many-to-one name="defaultValue" column="DEFAULT_VALUE"
    unique="true" not-null="false"/>
</joined-subclass>

</joined-subclass>

<joined-subclass name="edu.upc.gessi.rptool.domain.metrics.
MetricSet" lazy="false" table="METRIC_SET">
  <key column="ID"/>
  <many-to-one name="simple" lazy="false" column="NAME_SIMPLE" not-
    null="true"/>
</joined-subclass>

</joined-subclass>

</class>

</hibernate-mapping>

```


Els atributs *id* de totes les classes que el tenen es generen de forma automàtica a la Base de Dades, despreocupant al programador d'aquesta tasca. Els atributs que són de tipus simple (enters, floats, strings, dates, etc.) es mapegen d'una forma molt simple mitjançant el tag *property*. En quant als atributs que són pròpiament objectes definits pel programador o conjunt d'aquests, cal destacar:

- **Relacions many-to-one.** Les relacions many-to-one, que correspon a tenir un atribut que és un objecte dins d'una classe, es mapegen d'una forma molt simple. Ho explicarem amb un exemple. En el cas d'un *Parameter*, aquest sempre té una *Metric* associada, però una *Metric* pot estar associada a més d'un *Parameter*. Per tant, això és una relació many-to-one en la navegabilitat de *Parameter* a *Metric*. El que farem és, doncs, al mapeig de la classe *Parameter*, afegir:

```
<many-to-one name="myMetric" column="MY_METRIC" not-null="true"/>
```

Tenint en compte que l'atribut a la classe *Parameter* de tipus *Metric* és *myMetric*.

- **Relacions one-to-many.** Les relacions one-to-many, que correspon a tenir un atribut que és un conjunt d'objectes dins d'una classe, es mapegen tal com segueix. Ho explicarem mitjançant un exemple. Un *PatternItem* pot tenir zero o més *Parameter* associats, però un *Parameter* només pot pertànyer a un *PatternItem*. Per tant, això és una relació one-to-many en la navegabilitat de *PatternItem* a *Parameter*. El que farem és, doncs, al mapeig de la classe *PatternItem* afegir:

```
<set name="myParameters" lazy="false">
  <key column="MY_PATTERN_ITEM"/>
  <one-to-many class="edu.upc.gessi.rptool.domain.
    requirementpatterns.Parameter"/>
</set>
```

Tenint en compte que l'atribut a la classe *PatternItem* que és un conjunt de *Parameter* és *myParameters*.

8.3. Modificació de la Base de Dades existent

Degut al nou atribut *year* de la subclasse *Time Point* de la classe mètrica, va ser necessari modificar la Base de Dades per tal d'afegir una nova columna a la taula de les mètriques de tipus time point que representés aquest nou atribut.

Per tal de no perdre el contingut actual de la Base de Dades, en comptes de crear una nova Base de Dades que ja tingués aquesta nova columna a la taula, el que es va fer és afegir la columna a la taula ja existent. Per això, el que es va fer és, mitjançant el programa SQuirreL-SQL Client (per més informació sobre el programa veure l'apartat 3.3), connectar-se a la Base de Dades de patrons i mitjançant una comanda SQL afegir la nova columna a la taula en qüestió.

Tenint en compte que la taula de les mètriques de tipus time point és TIME_POINT_METRIC i que tal com s'ha establert al fitxer de mapeig d'Hibernate la nova columna és diu YEAR_1, la comanda SQL utilitzada va ser:

```
ALTER TABLE TIME_POINT_METRIC ADD COLUMN "YEAR_1" int
```

9. Segona versió de RPToolManager

1. Introducció.....	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2^a versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1 ^a versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
9.1. Modificació del codi font de l'aplicació.....	76
9.2. Proves del Software.....	76
PART 2: Desenvolupament de la 1^a versió de l'aplic. RPProjectManager.	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
ANNEX 1: Modificacions al codi font de RPToolManager	265

9.1. Modificació del codi font de l'aplicació

Després de realitzar l'anàlisi de la primera versió de l'aplicació i de modificar el model conceptual, es va arribar a la fase de modificació del codi font de l'aplicació per tal de corregir els errors i afegir les mancances i noves funcionalitats detectades durant l'anàlisi.

A l'annex 1 d'aquest document es pot trobar la taula on s'indica, per cada error/mancança/funcionalitat, com es va acabar solucionant i implementant.

Cal tenir en compte, però, que aquesta fase del projecte va ser la que més temps va portar, una mica més d'un mes i mig, (per més informació sobre la planificació veure l'apartat 20, Planificació i Execució del Projecte). A part del fet de que hi havia bastants errors, mancances i noves funcionalitats a corregir i afegir, això va ser conseqüència directa de dos motius principals:

1. Per una banda, no hi havia cap documentació escrita referent al disseny de l'aplicació, ni tan sols el propi codi font estava documentat, amb la qual cosa el fet de localitzar d'on provenia un error o a on s'havia de corregir era molt complicat, sobretot inicialment.
2. No es va poder adaptar l'entorn gràfic de l'aplicació per a que aquest pogués ser utilitzat per l'editor de finestres d'Eclipse. Tot i que en les fases d'inici de l'aplicació un dels seus desenvolupadors va dir que l'entorn gràfic si que es va realitzar amb l'editor de finestres, la realitat va ser que el codi font no s'adaptava a l'editor, probablement degut a que després es van realitzar canvis a mà a la part gràfica, desestructurant així el codi per tal que fos llegible per l'editor. Això va comportar que cada cop que s'havia de modificar alguna finestra de l'aplicació (per afegir o eliminar qualsevol component), els canvis portaven molt de temps ja que era costós que després del canvi la finestra tornés a veure's correctament quan s'executava l'aplicació.

9.2. Proves del Software

Aquest artefacte té com a objectiu la comprovació, de manera formal, del bon funcionament de totes les modificacions que s'han realitzat a l'eina. A tal efecte, s'ha comprovat la resposta de l'aplicació després de la implementació de les modificacions tant per la visualització de la interfície (que es mostressin bé els arbres, les finestres taronges i els camps dels diàlegs) com per la informació emmagatzemada a la Base de Dades.

A continuació es mostren les diferents taules de comprovació pels errors, mancances i noves funcionalitats detectades durant l'anàlisi de l'aplicació.

Errors a nivell de domini

Error	Visualització	BBDD	Error	Visualització	BBDD
D1	✓	✓	D5	✓	✓
D2	✓	✓	D6	✓	✓
D3	✓	✓	D7	✓	✓
D4	✓	✓	D8	✓	✓

Taula 2: Resultats de les proves del software, errors a nivell de domini

Errors a nivell d'interfície

Error	Visualització	BBDD	Error	Visualització	BBDD
I1	✓	---	I38	✓	✓
I2	✓	---	I39	✓	✓
I3	✓	---	I40	✓	✓
I4	✓	---	I41	✓	✓
I5	✓	✓	I42	✓	✓
I6	✓	✓	I43	✓	✓
I7	✓	✓	I44	✓	✓
I8	✓	✓	I45	✓	✓
I9	✓	✓	I46	✓	---
I10	✓	✓	I47	✓	---
I11	✓	✓	I48	✓	---
I12	✓	✓	I49	✓	✓
I13	✓	✓	I50	✓	✓
I14	✓	✓	I51	✓	✓
I15	✓	---	I52	✓	✓
I16	✓	✓	I53	✓	✓
I17	✓	✓	I54	✓	✓
I18	✓	✓	I55	✓	✓
I19	✓	✓	I56	✓	✓
I20	✓	---	I57	✓	✓
I21	✓	✓	I58	✓	✓
I22	✓	✓	I59	✓	✓
I23	✓	✓	I60	✓	✓
I24	✓	✓	I61	✓	✓
I25	✓	✓	I62	✓	---
I26	✓	---	I63	✓	---
I27	✓	✓	I64	✓	---
I28	✓	✓	I65	✓	✓
I29	✓	✓	I66	✓	✓
I30	✓	✓	I67	✓	✓
I31	✓	✓	I68	✓	✓
I32	✓	✓	I69	✓	✓
I33	✓	✓	I70	✓	✓
I34	✓	✓	I71	✓	✓
I35	✓	✓	I72	✓	✓

I36	✓	✓	I73	✓	✓
I37	✓	---	I74	✓	✓

Taula 3: Resultats de les proves del software, errors a nivell d'interfície

Mancances

Mancança	Visualització	BBDD	Mancança	Visualització	BBDD
M1	✓	---	M13	✓	✓
M2	✓	✓	M14	✓	---
M3	✓	✓	M15	✓	---
M4	✓	---	M16	✓	---
M5	✓	---	M17	✓	---
M6	✓	---	M18	✓	---
M7	✓	---	M19	✓	---
M8	✓	---	M20	✓	---
M9	✓	---	M21	✓	✓
M10	✓	---	M22	✓	---
M11	✓	---	M23	✓	---
M12	✓	---	M24	✓	---

Taula 4: Resultats de les proves del software, mancances

Noves funcionalitats

Funcionalitat	Visualització	BBDD
F1	✓	---
F2	✓	---
F3	✓	✓
F4	✓	✓

Taula 5: Resultats de les proves del software, noves funcionalitats

PART 2:
Desenvolupament
de la 1^a versió
de l'aplicació
RPPProjectManager

10. Metodologia: Àgil

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1 ^a versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager. 79	
10. Metodologia: Àgil	81
10.1. Introducció	82
10.2. Breu història.....	82
10.3. Els principis de la metodologia àgil.....	83
10.4. Per què les metodologies àgils funcionen?	83
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

10.1. Introducció

La programació àgil és una metodologia de desenvolupament de projectes en l'enginyeria del software que promou el desenvolupament a base d'iteracions de curta durada, en contraposició de projectes on el cicle de vida d'una iteració és molt més llarga.

Existeixen diversos mètodes de desenvolupament basats en els principis de la programació àgil, dels quals la majoria procuren minimitzar els riscos via el desenvolupament del software en períodes de temps molt curts. Aquests períodes o iteracions inclouen tots els artefactes d'un projecte software, això és: anàlisi de requeriments, disseny, codificació, testeig i documentació. En aquest tipus de metodologies una iteració no garanteix un producte funcional per a ser utilitzat sinó que l'objectiu és poder tenir disponible un release sense bugs, al final de cada iteració, que pugui ser mostrat al client i acceptat per part seva abans de continuar amb la iteració següent.

Les metodologies àgils emfatitzen la comunicació cara a cara més que no pas de documents escrits. També emfatitzen que treballar en el software és la primera mesura de progrés.

Combinat amb la preferència de comunicació face-to-face, aquest tipus de metodologies acostumen a produir menys documentació que les pesades. Aquest últim punt és la principal crítica que reben per part de seguidors més fermes en les metodologies pesades.

10.2. Breu història

La definició moderna de desenvolupament de software àgil ha anat evolucionant en la meitat dels 1990 com a part d'una reacció en contra de les metodologies pesades, com és per exemple la metodologia en cascada [WS07].

Els processos originats de l'ús de la metodologia en cascada es veien com burocràtics, lents, sense significat i inconsistents amb els mètodes amb que els enginyers de software realment desenvolupaven feina efectiva i productiva.

Inicialment les metodologies àgils eren anomenades “metodologies lleugeres” o “lightweight methods”. En el 2001, membres prominents d'aquesta comunitat van adoptar el nom de “metodologies àgils”. Posteriorment, alguns d'aquest membres van formar *The Agile Alliance*, una organització no lucrativa que promogué les metodologies àgils.

Metodologies similars o que compleixen els principis de les metodologies àgils van ser creades anteriorment. Entre elles destaquen el Scrum (1986), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, i DSDM (1995).

10.3. Els principis de la metodologia àgil

Com s'ha dit, les metodologies àgils promouen el desenvolupament evolutiu del programari durant tot el projecte per tal de reduir-ne el risc i donen més importància al programari que funciona, la comunicació, la cooperació i l'adaptació als canvis que a la documentació, els processos, els contractes i la planificació. L'any 2001, representants de diverses metodologies àgils van redactar l'*Agile Manifesto*, una declaració de principis en els quals es basen aquestes metodologies. Alguns d'aquests principis són [WEB Met. àgil]:

- Satisfacció del client per mitjà d'entregues ràpides i continuades de programari útil.
- El programari que funciona és la mesura principal del progrés.
- Fins i tot els canvis tardans en els requeriments són benvinguts.
- La conversa cara a cara és la millor forma de comunicació.
- Els projectes es construeixen al voltant de persones motivades, en les quals cal confiar.
- Atenció continuada a l'excel·lència tècnica i al bon disseny.
- Simplicitat.
- Equips autoorganitzats.
- Adaptació periòdica a circumstàncies canviants.

10.4. Per què les metodologies àgils funcionen?

Un dels grans debats creixents en l'enginyeria del software i en la indústria de les tecnologies de la informació en general és si les tècniques de desenvolupament de software àgils funcionen o no [WEB WhyAgileWorksFeedback]. Segons els defensors d'aquesta metodologia, així com també de diversos particulars o companyies que han instaurat satisfactòriament aquesta metodologia per desenvolupar els seus projectes, la resposta és Sí. Una de les principals raons que permet explicar el bon funcionament d'aquest tipus de metodologies és que es redueix el cicle de temps per obtenir feedback des de l'origen de la idea (o obtenció del requeriment) fins a la realització d'aquesta idea. Aquest fet no només minimitza el risc de malentesos, sinó que també redueix els costos d'errors realitzats en l'anàlisi de requeriments.

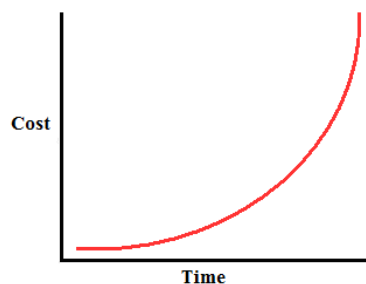


Figura 11: Corba de costos per canvis en un projecte

En la figura anterior es mostra la corba tradicional dels costos dels canvis. En aquest es veu clarament que com més tardà és el descobriment de que s'ha de realitzar un canvi més costa reparar-lo. El cost s'incrementa exponencialment. És per aquest motiu que cada cop més s'ha anat abandonant el desenvolupament de metodologies basades en fases de cascada, per altres metodologies iteratives o incrementals.

En la figura que es mostra a continuació es mostren els cicles de feedbacks en les tècniques de desenvolupament de software. Tècniques àgils, com Test Driven Design (TDD), Pair Programming i Agile Model Driven Development (AMDD) tenen cicles de feedback molt curts. Tècniques tradicionals com revisions, inspeccions i Big Requirements Up Front (BRUF) tenen cicles de feedback de l'ordre de setmanes o mesos, fent llavors els canvis amb un major risc i més cars.

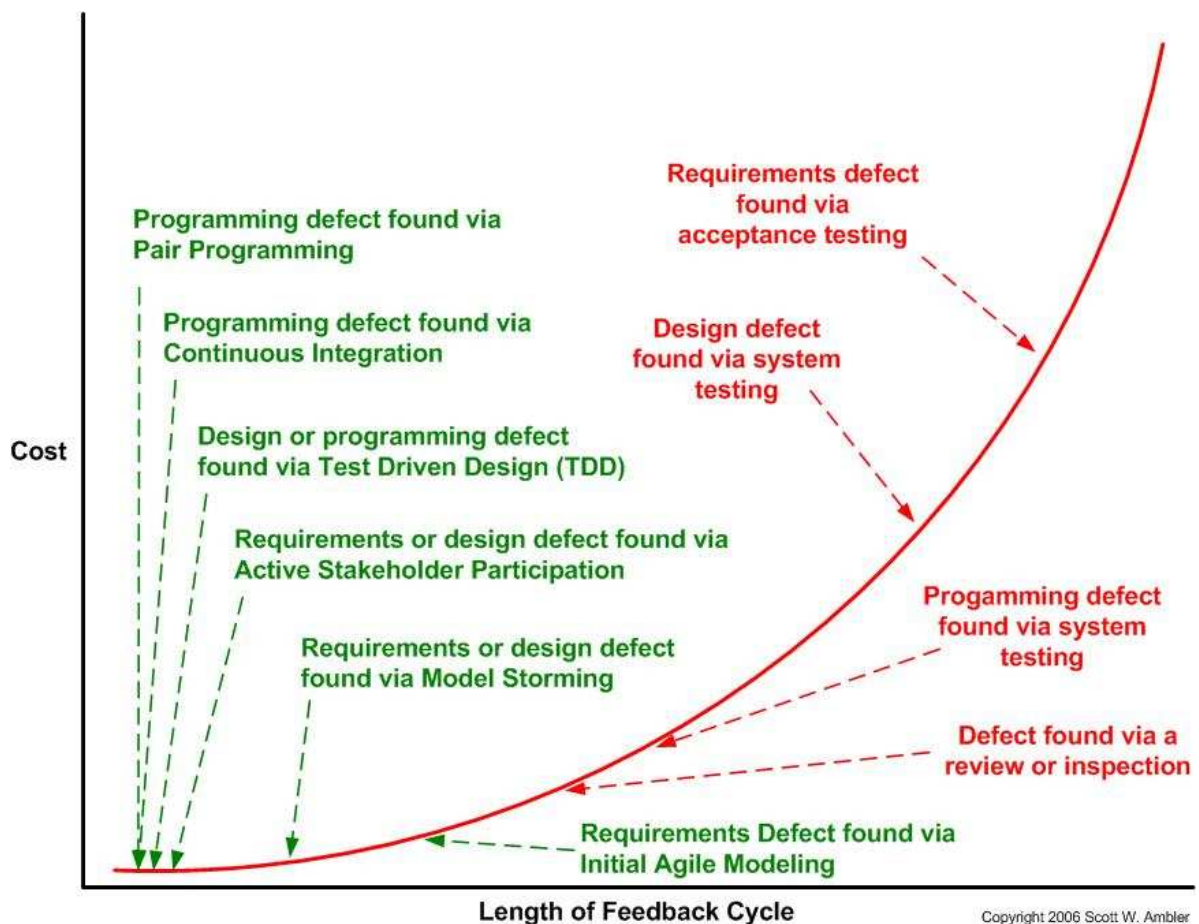


Figura 12: Comparativa del feedback obtingut entre diferents tècniques de metodologies de desenvolupament de software

11. Anàlisi de Requeriments

1.	Introducció	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1ª versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades.....	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
11.1.	<i>Propòsit de l'anàlisi de requeriments</i>	<i>86</i>
11.2.	<i>Abast del projecte</i>	<i>86</i>
11.2.1.	Funcionalitats que entren dins de l'abast del projecte	86
11.2.2.	Funcionalitats que queden fora de l'abast del projecte.....	86
11.3.	<i>Requeriments funcionals</i>	<i>87</i>
11.3.1.	Visualització de projectes i patrons.....	88
11.3.2.	Construcció de projectes.....	89
11.3.3.	Gestió de projectes	91
11.3.4.	Interoperabilitat.....	92
11.4.	<i>Requeriments no funcionals</i>	<i>92</i>
11.4.1.	Usability	93
11.4.2.	Reliability	93
11.4.3.	Performance.....	94
11.4.4.	Supportability.....	94
11.4.5.	Implementation Constraints.....	95
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work.....	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

11.1. Propòsit de l'anàlisi de requeriments

Els objectius principals d'aquest artefacte, l'anàlisi de requeriments, és l'obtenció de les condicions i capacitats que ha de satisfer l'aplicació per tal que les necessitats del client es vegin acomplertes per l'eina. En aquest capítol es tractarà de:

- Obtenir els requeriments de l'usuari pel que fa a l'eina.
- Definir de manera inequívoca cadascun d'aquests requeriments mitjançant una descripció completa de les necessitats de l'usuari en cada aspecte
- Validar els requeriments obtinguts i que s'han d'acomplir en el sistema desenvolupat mitjançant l'aprovació de la client Carme Quer.
- Construir un catàleg de requeriments funcionals que serveixin com a pas previ per a la definició dels casos d'ús.

11.2. Abast del projecte

11.2.1. Funcionalitats que entren dins de l'abast del projecte

- Visualitzar els diferents projectes i patrons que existeixen als repositoris.
- Constucció i gestió de projectes. Permetre crear/modificar/eliminar les diferents dades d'un projecte, que inclouen, a més dels atributs propis d'un projecte, els requisits associats a aquest. Per tant, l'aplicació també ha de permetre crear nous requisits (ja sigui un requisit nou o una instància d'un patró de requisit), modificar-los i eliminar-los i, a més, pels requisits que són instància d'un patró, ha de permetre donar valor als paràmetres del patró, en cas que en tingui.
- Generar el document pel call-for-tenders.

11.2.2. Funcionalitats que queden fora de l'abast del projecte

- Navegació seqüencial i automàtica pel catàleg de patrons de requisits. Tal com s'indica al mètode PABRE, la idea a grans trets és que durant el procés d'elicitació es recorri tot el catàleg de patrons i per cada patró, depenent de si interessa o no, instanciar-lo o no (per més informació sobre el mètode veure capítol 2.4). Ara mateix aquesta navegació s'ha de fer de forma manual. La idea

és que en una segona versió, sigui l'aplicació qui faci aquesta navegació i et proposi el següent patró que toca revisar.

- Estadístiques d'ús dels patrons de requisits.
- Tractar les dependències entre els requisits que són instància de patrons de requisits.

11.3. Requeriments funcionals

Els requeriments funcionals de l'aplicació s'han agrupat i categoritzat segons el tipus de funcionalitat que proporcionen. En aquest sentit s'han classificat els requeriments d'aquest tipus en quatre grups:

- Visualització de projectes i patrons
- Construcció de projectes
- Gestió de projectes
- Interoperabilitat

Format

El format en que s'escriuen els requeriments està inspirat en les fitxes de requeriments de la metodologia d'obtenció i elicitació de requeriments de Volere [WEB Reqs. Volere]. Tot i estar inspirat en Volere, degut a que no es podria dedicar tantes pàgines només a la descripció de requeriments, la plantilla resultant és una simplificació d'aquest.

Per cada requeriment es mostra:

- Identificador
- Nom del requeriment
- Descripció i justificació del requeriment
- Fit criteria

Req. #:	Id
Name:	
Nom del requeriment	
Description & Rationale:	
Descripció i justificació del requeriment	
Fit Criteria:	
Condicó a partir de la qual es pot afirmar inequívocament que el requeriment s'ha assolit	

11.3.1. Visualització de projectes i patrons

Els requeriments funcionals que es citen a continuació descriuen els diferents tipus de visualització d'objectes que l'eina ha de permetre:

- Visualitzar tots els projectes en ordre alfabètic.
- Visualitzar els atributs d'un projecte.
- Visualitzar tota la informació d'un projecte.
- Visualitzar tots els patrons segons el classification schema escollit.
- Visualitzar els atributs d'un element d'un patró.
- Visualitzar tota la informació d'un patró.

Req. #:	1.1
Name:	
Visualitzar tots els projectes en ordre alfabètic	
Description & Rationale:	
S'ha de poder permetre visualitzar la llista en ordre alfabètic de tots els noms dels projectes que existeixen al repositori.	
Fit Criteria:	
La llista en ordre alfabètic de tots els noms dels projectes que existeixen al repositori es visualitza en un fragment de la pantalla de l'aplicació.	

Req. #:	1.2
Name:	
Visualitzar els atributs d'un projecte	
Description & Rationale:	
S'ha de poder permetre visualitzar els atributs de qualsevol projecte.	
Fit Criteria:	
Els atributs d'un projecte seleccionat són visualitzats en un fragment de la pantalla de l'aplicació.	

Req. #:	1.3
Name:	
Visualitzar tota la informació d'un projecte	
Description & Rationale:	
S'ha de poder permetre visualitzar tota la informació de qualsevol projecte.	
Fit Criteria:	
Tota la informació d'un projecte seleccionat es visualitza en una nova finestra de l'aplicació.	

Req. #:	1.4
Name:	
Visualitzar tots els patrons segons el classification schema escollit	
Description & Rationale:	
S'ha de poder permetre visualitzar la llista de tots els patrons i els seus elements (forms, extended i fixed parts) que existeixen al repositori segons el classification schema seleccionat.	
Fit Criteria:	
La llista de tots els patrons i els seus elements que existeixen al repositori es visualitza en un fragment de la pantalla de l'aplicació segons el classification schema seleccionat.	

Req. #:	1.5
Name:	
Visualitzar els atributs d'un element d'un patró	
Description & Rationale:	
S'ha de poder permetre visualitzar els atributs de qualsevol element d'un patró (el patró en si mateix, o qualsevol dels seus forms, fixed o extended parts).	
Fit Criteria:	
Els atributs d'un element d'un patró seleccionat són visualitzats en un fragment de la pantalla de l'aplicació.	

Req. #:	1.6
Name:	
Visualitzar tota la informació d'un patró	
Description & Rationale:	
S'ha de poder permetre visualitzar tota la informació de qualsevol patró.	
Fit Criteria:	
Tota la informació d'un patró seleccionat es visualitza en una nova finestra de l'aplicació.	

11.3.2. Construcció de projectes

Els següents requeriments fan referència a totes aquelles funcionalitats que ha de satisfer l'aplicació per tal de poder construir projectes de manera consistent i efectiva. En concret:

- Crear un projecte.
- Editar un projecte.
- Crear un requisit en un projecte.
- Editar un requisit d'un projecte.
- Eliminar un requisit d'un projecte.

Req. #:	2.1
Name:	
Crear un projecte	
Description & Rationale:	
S'ha de poder permetre la creació de nous projectes.	
Fit Criteria:	
Es poden crear satisfactòriament nous projectes mitjançant l'aplicació.	

Req. #:	2.2
Name:	
Editar un projecte	
Description & Rationale:	
Els atributs d'un projecte, com són el nom o la descripció, s'han de poder editar.	
Fit Criteria:	
Tots els atributs de tots els projectes es poden editar des de l'aplicació.	

Req. #:	2.3
Name:	
Crear un requisit en un projecte	
Description & Rationale:	
S'ha de poder permetre la creació de qualsevol tipus de requisits sobre un projecte.	
Fit Criteria:	
Sobre un projecte concret es poden crear satisfactòriament els següents tipus de requisits: <ul style="list-style-type: none"> • Un requisit que és una instància d'un patró. • Un requisit que té un patró associat. • Un requisit "nou" (que no és ni una instància d'un patró ni té cap patró associat). 	

Req. #:	2.4
Name:	
Editar un requisit d'un projecte	
Description & Rationale:	
S'ha de permetre l'edició del nom i altres atributs dels requisits d'un projecte, així com també canviar-los de tipus.	
Fit Criteria:	
Es poden modificar el nom, el question text i el form text de tots els requisits de tots els projectes. A més, per qualsevol requisit, si aquest és: <ul style="list-style-type: none"> • Un requisit que és una instància d'un patró, es pot canviar també el patró que instancia, qualsevol valor d'un dels seus paràmetres (si és que en té) i canviar-li el tipus a un requisit amb un patró associat. També s'ha de poder fer que el seu question text i form text tornin a ser 	

els mateixos que els del patró que instancia.

- Un requisit amb un patró associat, es pot canviar el patró que té associat i canviar-li el tipus, passant a ser un requisit de tipus “nou” (que no és ni una instància d’un patró ni té cap patró associat).
- Un requisit “nou”, es pot canviar els tipus a un requisit amb un patró associat.

Req. #:	2.5
Name:	
Eliminar un requisit d’un projecte	
Description & Rationale:	
S’ha de poder eliminar un requisit d’un projecte.	
Fit Criteria:	
Qualsevol requisit de qualsevol projecte ha de quedar completament eliminat.	

11.3.3. Gestió de projectes

Els següents requeriments fan referència al tractament de projectes com a entitats de l’aplicació. En aquest sentit s’han identificat els següents requeriments funcionals:

- Copiar un projecte.
- Eliminar un projecte.

Req. #:	3.1
Name:	
Copiar un projecte	
Description & Rationale:	
S’ha de permetre copiar un projecte a un altre projecte amb els mateixos atributs i requisits que l’original, però amb diferent nom.	
Fit Criteria:	
El projecte copiat té exactament els mateixos atributs i requisits que el projecte original i l’única diferència està en el nom del projecte, que obligatòriament ha de ser diferent (no poden existir dos projectes amb el mateix nom).	

Req. #:	3.2
Name:	
Eliminar un projecte	
Description & Rationale:	
S’ha de poder eliminar un projecte.	
Fit Criteria:	
El projecte ha quedat completament suprimit de l’aplicació així com també tots els seus requisits.	

11.3.4. Interoperabilitat

Els requeriments d'interoperabilitat tracten sobre les funcionalitats d'exportació de projectes a documents amb extensió pdf. Més concretament:

- Generar pdf amb tota la informació d'un projecte.
- Generar pdf amb el document call-for-tenders d'un projecte.

Req. #:	4.1
Name:	
Generar pdf amb tota la informació d'un projecte	
Description & Rationale:	
S'ha de poder permetre generar un document pdf amb tota la informació d'un projecte.	
Fit Criteria:	
Per qualsevol projecte, es pot generar el pdf que es correspon amb tota la informació del projecte.	

Req. #:	4.2
Name:	
Generar pdf amb el document call-for-tenders d'un projecte	
Description & Rationale:	
S'ha de poder permetre generar un document pdf amb la informació del document de call-for-tenders d'un projecte.	
Fit Criteria:	
Per qualsevol projecte, es pot generar el pdf que es correspon amb la informació del document de call-for-tenders del projecte.	

11.4. Requeriments no funcionals

Per l'obtenció dels requeriments no funcionals s'ha decidit utilitzar la categorització del conegut com a FURPS+ on els requeriments estan agrupats en [WEB FURPS+]:

- Funcionals (descrits en el capítol anterior)
- Usability
- Reliability
- Performance
- Security
- Supportability
- Implementation Constraints

11.4.1. Usability

[RNF 1.1] Speed of Use

Les funcionalitats de l'aplicació s'han d'executar satisfactòriament de la forma més ràpida possible. S'entén que no és necessari processar cap gran volum d'informació, ni tampoc és necessari cap algorisme el cost del qual sigui exponencial, per tant, es pot definir com a requeriment assolible que el cost en temps de cap funcionalitat de l'aplicació superi, en segons, els $0,04 * \text{numObjectesImplicats}$.

[RNF 1.2] Required User Ability

L'aplicació *RPPProjectManager* ha d'estar dissenyada per ser utilitzada per usuaris experts, o que almenys estiguin familiaritzats, amb el procés d'elicitació de requisits i el concepte de patrons de requisits.

[RNF 1.3] Learnability

Pensat per a aquests tipus d'usuaris, l'usuari d'aquesta eina ha de poder aprendre fàcilment totes les funcionalitats de l'aplicació en menys d'un dia hàbil. La majoria de les funcionalitats s'han de poder ensenyar en un curs/demostració de 3 hores de durada.

[RNF 1.4] Training Material

L'aplicació resultant haurà d'incloure un mínim de dos projectes, com a base per a poder visualitzar i treballar amb exemples.

11.4.2. Reliability

[RNF 2.1] Maximum Down Time

El màxim temps que el sistema pot restar inactiu per una fallada (per exemple, el SGBD no respon a les peticions per un overloading generat per excés de peticions), ha de ser menor a 1 minut, que és el temps per defecte del SQL per llançar un `TimeoutException`.

[RNF 2.2] Maximum Known Bugs

Malgrat que la voluntat de tot client i desenvolupador és que tota aplicació sigui 100% error-free, sembla inevitable que tot programa hagi de contenir certa quantitat de minor-bugs no detectats en la fase de desenvolupament. Per a especificar el següent requeriment s'usarà l'habitual bugs-allowed per thousand lines of code. En el cas de *RPPProjectManager*, s'ha acordat que per les prestacions de l'aplicació, serien de 0,5 bugs per cada 1000 línies de codi.

11.4.3. Performance

[RNF 3.1] Response Time

El temps de resposta per a qualsevol funcionalitat de l'aplicació, al no requerir d'algorismes amb un cost massa elevat en temps, ha de ser menor de $0,04 * \text{numObjectesImplicats}$ segons per a les funcionalitats que involucren a un projecte sencer (copiar un projecte, generar documents, etc.) i menor de 1 segon per a les funcionalitats que involucren un element del projecte (afegir un requisit, editar el valor d'un paràmetre, etc.).

[RNF 3.2] Resource Usage

L'aplicació encesa utilitzarà menys de 100 MB de memòria RAM. El cost d'espai en disc de l'aplicació (executable, imatges, etc), descomptant la informació que l'usuari pugui emmagatzemar a la BBDD, ha de ser inferior a 50 MB. El cost en temps de CPU de l'aplicació ha de ser inferior a 30 segons després d'un ús de l'eina d'1 hora.

[RNF 3.3] Security

En aquest sentit, no es requereix l'ús de contrasenyes ni és necessari impedir que un usuari pugui accedir a les dades d'un projecte creat per un altre usuari.

11.4.4. Supportability

[RNF 4.1] Ease of Installation

L'aplicació s'ha de poder instal·lar sense dificultats, tant per a usuaris com per a desenvolupadors. La instal·lació ha de comportar, com a molt, les tasques de copiar els fitxers i executar algun script.

[RNF 4.2] Planned Maintenance

El manteniment de l'aplicació és una qüestió prioritària en l'aplicació. Per aquest motiu la bona documentació i extensibilitat són requeriments prioritaris que ha de satisfer el programa.

[RNF 4.3] Ease of Configuration

Cap dels paràmetres interns de l'aplicació han de requerir una configuració diferent a la que porta per defecte l'aplicació. En aquest sentit, doncs, no s'ha de demanar cap tipus de configuració addicional per l'ús de l'eina.

11.4.5. Implementation Constraints

[RNF 5.1] Languages

El llenguatge de programació a utilitzar serà el JAVA versió 1.6. La interfície gràfica haurà d'usar les llibreries SWING, AWT.

[RNF 5.2] Operating Systems

L'aplicació ha de ser usable en Microsoft Windows XP o superior. Al ser programat en JAVA, la seva compatibilitat amb Linux i altres sistemes operatius és assolible, però no és una necessitat imperant per la client.

[RNF 5.3] Standards

Totes les funcions han d'estar documentades utilitzant el JavaDoc. Tant si són públiques, protected o privades

[RNF 5.4] System Interfaces

La interfície de l'aplicació i els diàlegs, així com també els menús, han de ser similars a les aplicacions tradicionals del sistema operatiu en que s'executi.

[RNF 5.5] Legacy Systems

L'aplicació ha de tenir una llicència lliure de tipus compatible amb EPL (Eclipse Public License) ja que es tracta d'una aplicació desenvolupada amb aquest IDE.

12. Model de Casos d'ús

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
12.1. Introducció	98
12.1.1. Paquets dels casos d'ús.....	98
12.1.2. Herència en els casos d'ús (UML 2.2)	98
12.1.3. Actors implicats en els casos d'ús	99
12.2. Diagrames de Casos d'ús	100
12.2.1. Visualització de projectes i patrons.....	100
12.2.2. Construcció de projectes.....	101
12.2.3. Gestió de projectes	102
12.2.4. Interoperabilitat.....	102
12.3. Descripció dels Casos d'ús	103
12.3.1. Visualització de projectes i patrons.....	104
12.3.2. Construcció de projectes.....	108
12.3.3. Gestió de projectes	121
12.3.4. Interoperabilitat.....	123
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager	265

12.1. Introducció

Els casos d'ús són un artefacte imprescindible en qualsevol projecte software. Els casos d'ús són un mecanisme per ajudar a mantenir d'una manera simple i comprensible els requeriments funcionals que complirà el producte, així com de quina forma ho farà, tant pel desenvolupador com pels stakeholders.

12.1.1. Paquets dels casos d'ús

Els casos d'ús s'han dividit en paquets per facilitar la seva lectura i tractament. Els paquets resultants coincideixen amb la categorització dels requeriments funcionals associats als casos d'ús, de manera que es facilita l'accés a un cas d'ús a partir del seu requeriment associat.

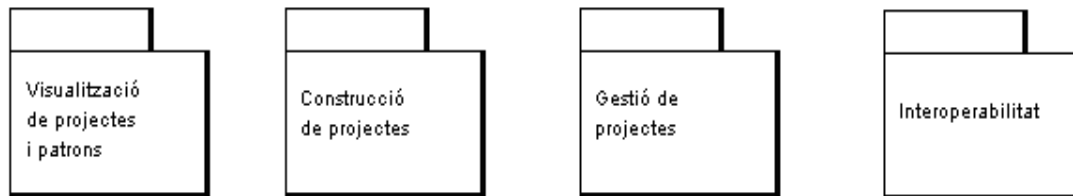


Figura 13: Paquets dels casos d'ús

12.1.2. Herència en els casos d'ús (UML 2.2)

En els casos d'ús implicats, n'hi ha alguns que s'han descrit com a subcasos d'ús (o herència) d'un cas d'ús major. Aquesta característica d'heretar propietats dels casos d'ús és sintàcticament i semànticament vàlida des de l'UML 2.0. En aquest sentit, s'ha utilitzat l'UML 2.2, la definició del qual es pot consultar a [WEB UML2.2]. En l'apartat dels casos d'ús es descriu com es representa l'herència en els diagrames de casos d'ús que es pot trobar a la referència anterior però no posa cap restricció en la descripció dels casos d'ús i deixa llibertat al desenvolupador en aquest sentit. Aquesta decisió de l'OMG (Object Management Group) ve presa partint de la base de que els casos d'ús es descriuen normalment en llenguatge natural i de maneres molt diverses i que, per tant, no es pot posar una semàntica a l'herència dins de la descripció dels casos d'ús.

12.1.3. Actors implicats en els casos d'ús

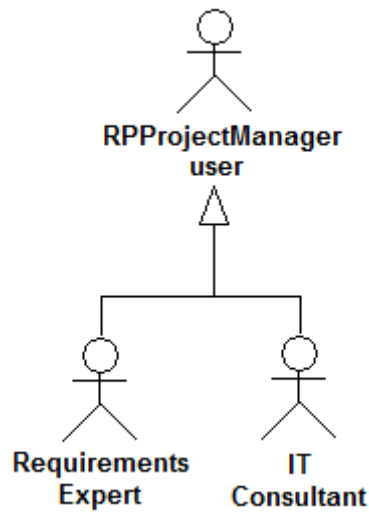


Figura 14: Diagrama d'actors

- **RPPProjectManager user.** Representa la generalització de qualsevol tipus d'actor que utilitza l'aplicació. Els casos d'ús al que aquest té accés són accessibles a tots els seus subactors.
- **Requirements Expert.** Representa l'usuari que utilitza l'aplicació per consultar l'ús dels patrons de requisits en els diferents projectes.
- **IT Consultant.** Representa l'usuari que utilitza l'aplicació per crear i modificar projectes, i també gestionar-los.

12.2. Diagrames de Casos d'ús

12.2.1. Visualització de projectes i patrons

Aquests casos d'ús es corresponen amb tots aquells que tenen a veure amb la visualització de la informació de projectes i patrons a l'aplicació.

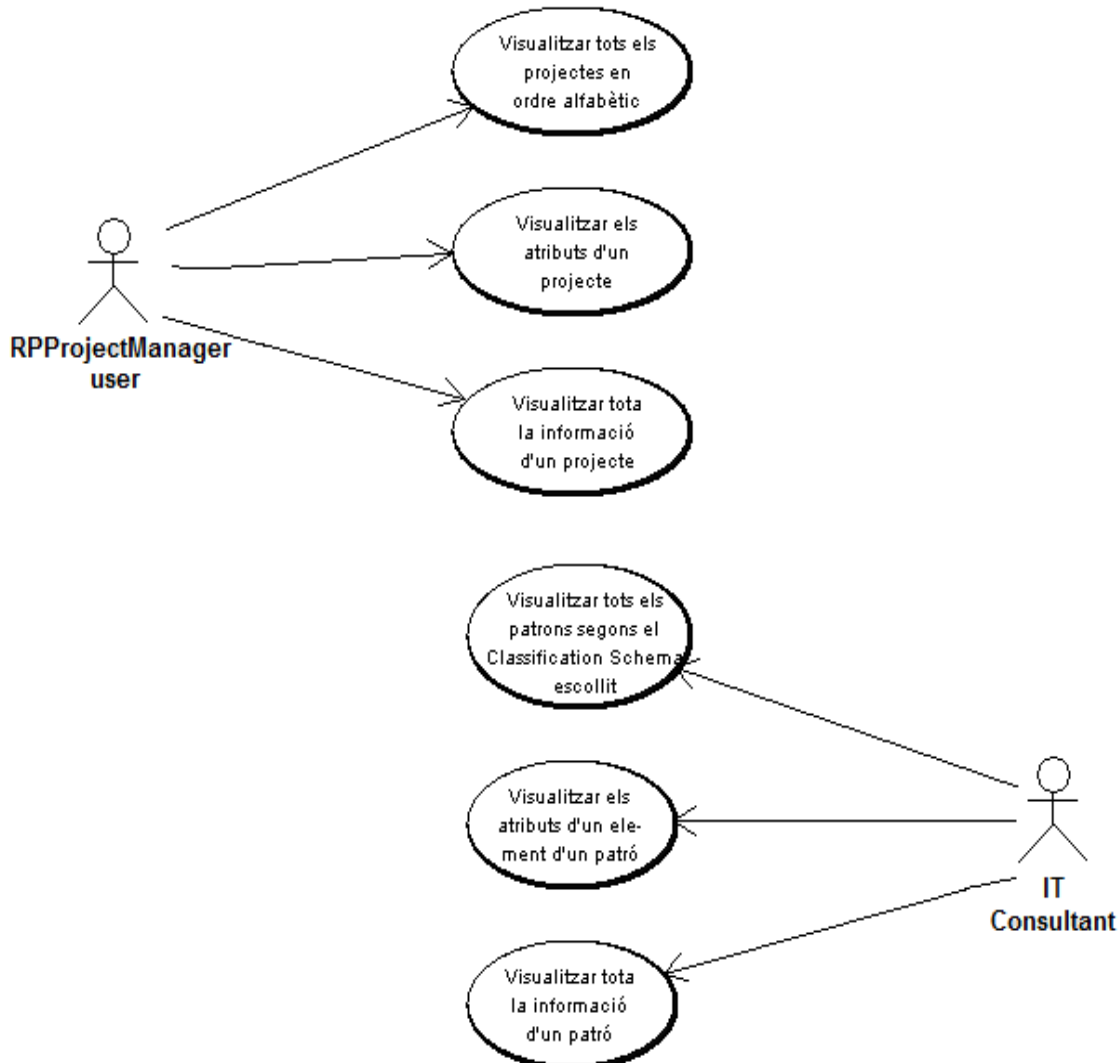


Figura 15: Diagrama de casos d'ús sobre *Visualització de projectes i patrons*

12.2.2. Construcció de projectes

Els següents casos d'ús es corresponen amb tot allò que fa referència a la construcció de projectes, ja sigui per crear-ne de nous o editar els atributs d'un existent, o per afegir, editar o eliminar els requisits d'un projecte.

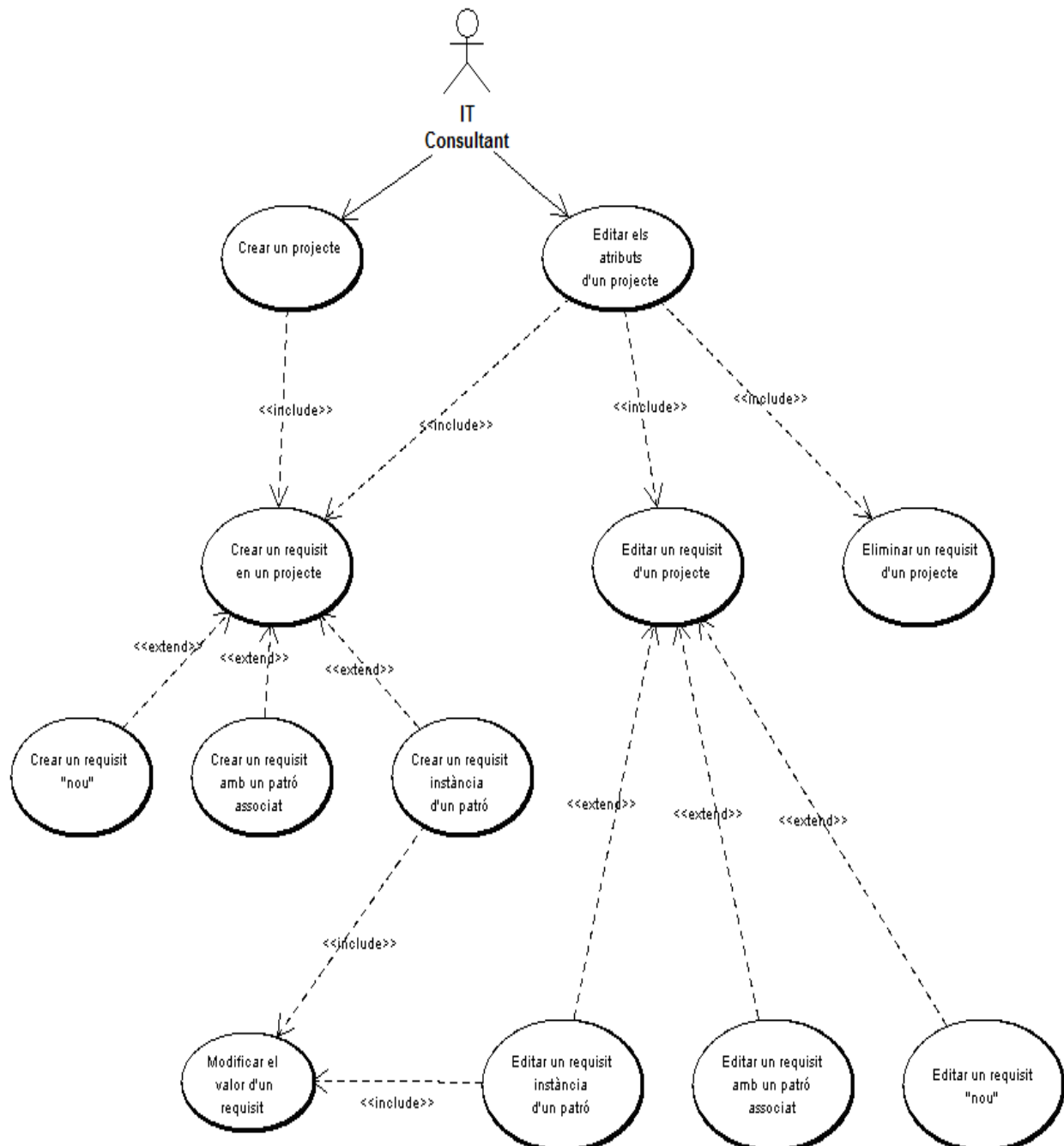


Figura 16: Diagrama de casos d'ús sobre *Construcció de projectes*

12.2.3. Gestió de projectes

Aquests són els casos d'ús referents a la gestió de projectes.

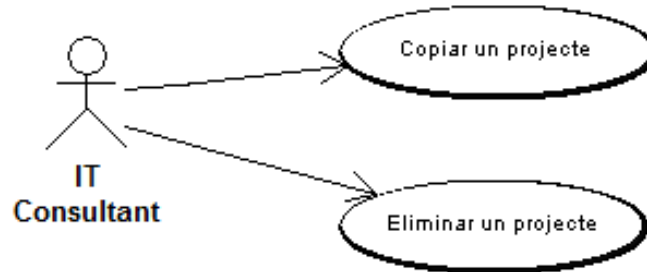


Figura 17: Diagrama de casos d'ús sobre *Gestió de projectes*

12.2.4. Interoperabilitat

Aquest casos d'ús són els que satisfan els requeriments funcionals d'interoperabilitat: generar un pdf amb tota la informació del projecte i generar un pdf amb el document call-for-tenders d'un projecte.

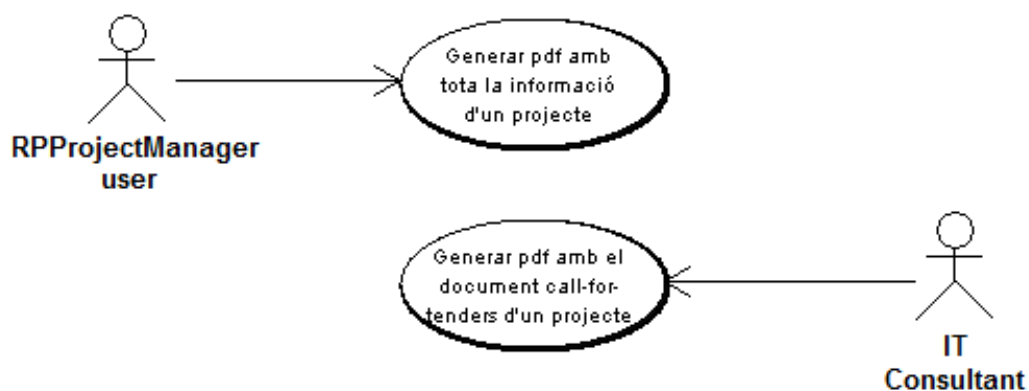


Figura 18: Diagrama de casos d'ús sobre *Interoperabilitat*

12.3. Descripció dels Casos d'ús

Per la descripció dels casos d'ús s'usarà la següent plantilla. En aquesta es mostra l'identificador, el nom i la descripció del cas d'ús. També s'indica el requeriment funcional que cobreix i els actors involucrats. Per últim, apareix el curs típic d'esdeveniments i els seus cursos alternatius.

Cas d'ús #	id
Nom:	
Nom del cas d'ús	
Descripció:	
Descripció del cas d'ús	
Precondició:	
Precondició a complir	
Req. coberts:	Req. F. #
Actors:	IT Consultant / Requirements expert
Curs típic d'esdeveniments	
Actor	Sistema
1. Acció de l'actor	
	2. Resposta del sistema
Cursos Alternatius	
Cursos Alternatius	

12.3.1. Visualització de projectes i patrons

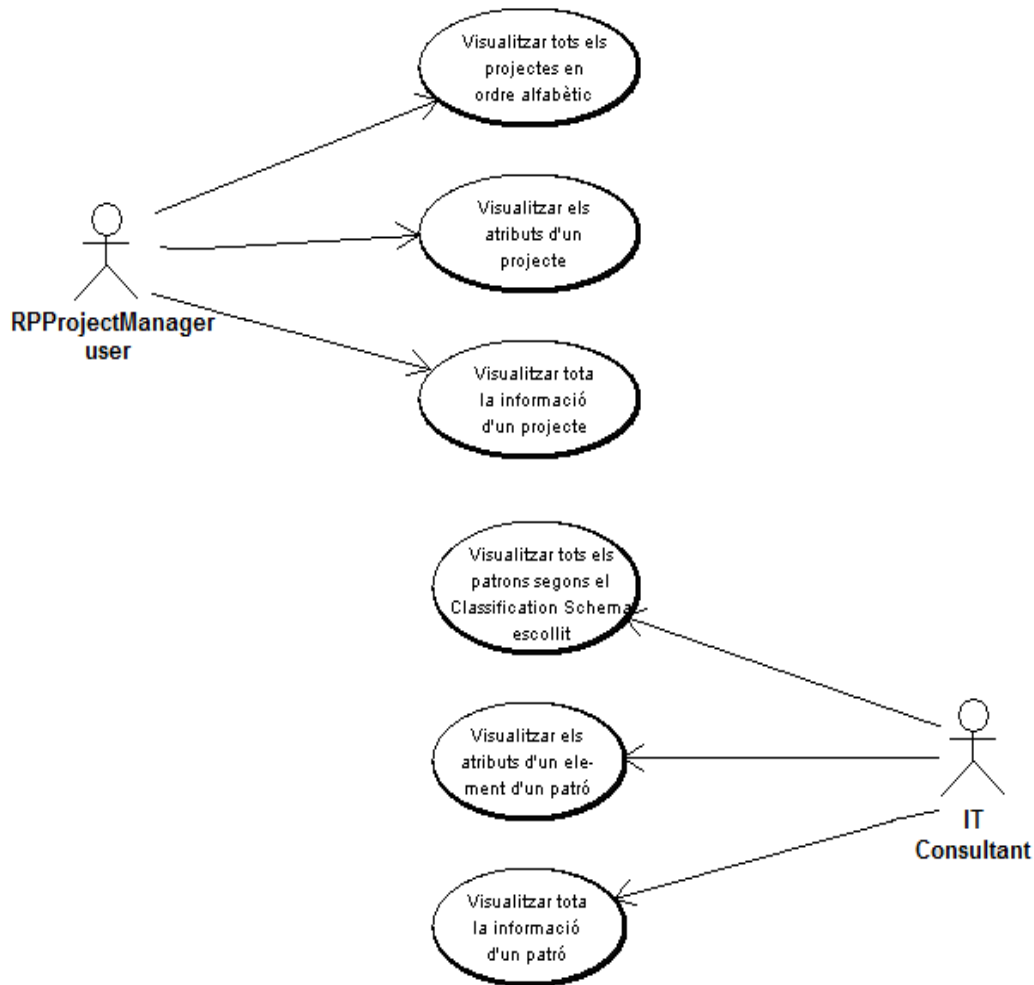


Figura 15: Diagrama de casos d'ús sobre *Visualització de projectes i patrons*

Cas d'ús #	1.1
Nom:	
Visualitzar tots els projectes en ordre alfabètic	
Descripció:	
Es recuperen tots els projectes que existeixen al repositori i es mostra una llista amb els seus noms ordenats alfabèticament.	
Precondició:	

Req. coberts:	RF 1.1
Actors:	IT Consultant, Requirements expert
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari executa l'aplicació o prem el botó per obrir la finestra de projectes.	

	2. El sistema carrega tots els projectes i els mostra en una part de la finestra de l'aplicació ordenats alfabèticament.
Cursos Alternatius	

Cas d'ús #	1.2
Nom:	
Visualitzar els atributs d'un projecte	
Descripció:	
Es visualitzen els atributs d'un projecte.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte del qual l'usuari vol veure els seus atributs.	
Req. coberts:	RF 1.2
Actors:	IT Consultant, Requirements expert
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un projecte per veure els seus atributs.	
	2. El sistema mostra els atributs del projecte.
Cursos Alternatius	

Cas d'ús #	1.3
Nom:	
Visualitzar tota la informació d'un projecte	
Descripció:	
Es visualitza tota la informació d'un projecte.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte del qual l'usuari vol veure tota la informació.	
Req. coberts:	RF 1.3
Actors:	IT Consultant, Requirements expert
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un projecte per veure tota la seva informació.	
	2. El sistema mostra tota la informació del projecte.
Cursos Alternatius	

Cas d'ús #	1.4
Nom:	
Visualitzar tots els patrons segons el classification schema escollit	
Descripció:	
Es recuperen tots els patrons que existeixen al repositori i es mostra una llista amb els seus noms i tots els seus elements (forms, fixed i extended parts) ordenats segons el classification schema escollit.	
Precondició:	
La llista dels classification schema està carregada.	
Req. coberts:	RF 1.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari canvia el classification schema escollit per mostrar els patrons.	
	2. El sistema carrega tots els patrons i els seus elements i els mostra en una part de la finestra de l'aplicació ordenats segons el classification schema escollit.
Cursos Alternatius	

Cas d'ús #	1.5
Nom:	
Visualitzar els atributs d'un element d'un patró	
Descripció:	
Es visualitzen els atributs d'un element d'un patró (el patró en si mateix, o qualsevol dels seus forms, fixed o extended parts).	
Precondició:	
La llista dels patrons i dels seus elements està carregada. Existeix l'element del qual l'usuari vol veure els seus atributs.	
Req. coberts:	RF 1.5
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un patró o un dels seus elements per veure els seus atributs.	
	2. El sistema mostra els atributs de l'element seleccionat.
Cursos Alternatius	

Cas d'ús #	1.6
Nom:	
Visualitzar tota la informació d'un patró	
Descripció:	
Es visualitza tota la informació d'un patró.	
Precondició:	
La llista dels patrons i dels seus elements està carregada. Existeix el patró del qual l'usuari vol veure tota la informació.	
Req. coberts:	RF 1.6
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un patró per veure tota la seva informació.	
	2. El sistema mostra tota la informació del patró.
Cursos Alternatius	

12.3.2. Construcció de projectes

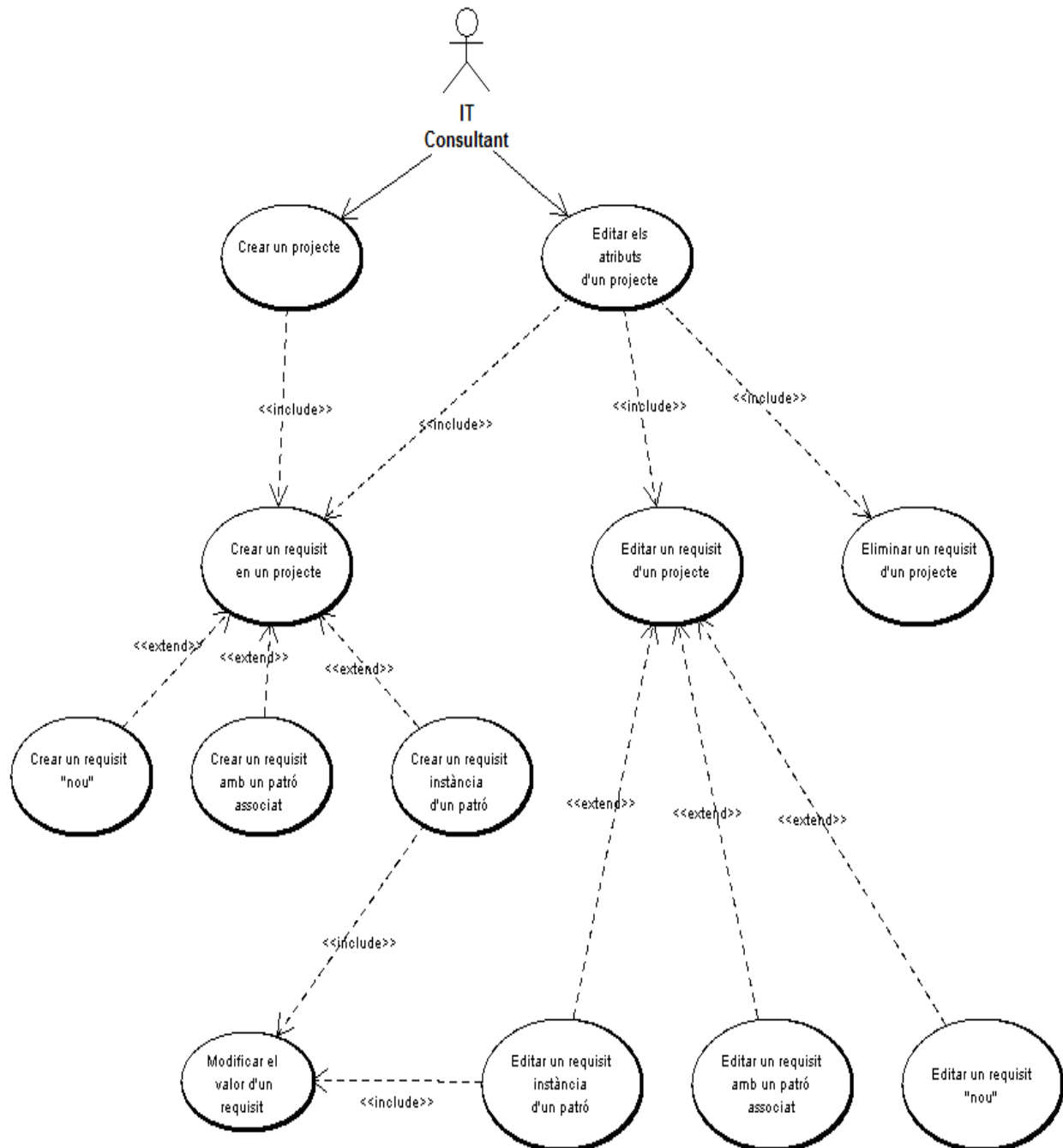


Figura 16: Diagrama de casos d'ús sobre *Construcció de projectes*

Cas d'ús #	2.1
Nom:	
Crear un projecte	
Descripció:	
S'afegeix un nou projecte al repositori.	
Precondició:	

Req. coberts:	RF 2.1, RF 2.3
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan l'usuari vol inserir un nou projecte.	
	2. El sistema carrega una finestra amb tots els patrons (i els seus elements) existents al repositori i la llista de tots els classification schema.
	3. El sistema mostra el diàleg de creació del projecte.
4. L'usuari omple els camps dels atributs del projecte.	
5. Opcionalment, l'usuari pot afegir requisits al projecte (veure cas d'ús 2.3).	
6. L'usuari accepta.	
	7. El sistema insereix el projecte al repositori.
Cursos Alternatius	
6a: el nom del projecte ja existeix al repositori	
	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació de projectes.

Cas d'ús #	2.2
Nom:	
Editar un projecte	
Descripció:	
S'editen els atributs d'un projecte.	
Precondició:	
La llista de projectes està carregada. El projecte a editar existeix.	
Req. coberts:	RF 2.2, RF 2.3, RF 2.4, RF 2.5
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona el projecte que vol editar i acciona l'opció corresponent.	
	2. El sistema carrega una finestra amb tots els patrons (i els seus elements) existents al repositori i la llista de tots els classification schema.
	3. El sistema mostra un diàleg des d'on es poden editar tots els atributs d'un projecte.
4. L'usuari edita els atributs del projecte que considera oportú.	
5. Opcionalment, l'usuari pot afegir requisits al projecte (veure cas d'ús 2.3).	
6. Opcionalment, l'usuari pot editar requisits del projecte (veure cas d'ús 2.4).	
7. Opcionalment, l'usuari pot eliminar requisits del projecte (veure cas d'ús 2.5).	
8. L'usuari accepta.	
	9. El sistema actualitza el projecte al repositori.
Cursos Alternatius	
8a: el nom del projecte ja existeix al repositori	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició de projectes.

El següent cas d'ús, el 2.3, té un comportament diferent i una interacció diferent segons el tipus de requisit que l'usuari vulgui afegir. És per això que es tracten com a subcasos d'aquest:

- 2.3.1: Crear un requisit “nou”.
- 2.3.2: Crear un requisit amb un patró associat.
- 2.3.3: Crear un requisit instància d'un patró.

Cas d'ús #	2.3
Nom:	
Crear un requisit en un projecte	
Descripció:	
Es crea un requisit i s'afegeix al projecte.	
Precondició:	
La llista dels patrons i dels seus elements està carregada. Existeix el projecte al que es vol afegir el nou requisit.	
Req. coberts:	RF 2.3
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan es vol afegir un requisit a un projecte.	
	2. El sistema mostra el diàleg per escollir el tipus de requisit a afegir.
3. L'usuari introdueix el nom del requisit i selecciona el tipus de requisit que vol afegir.	
	<i>[vegis subapartat corresponent segons el tipus de requisit escollit]</i>
Cursos Alternatius	

Cas d'ús #	2.3.1
Nom:	
Crear un requisit "nou"	
Descripció:	
Es crea un requisit de tipus "nou" i s'afegeix al projecte.	
Precondició:	
La llista dels patrons i dels seus elements està carregada. <i>(Extended from 2.3)</i> Existeix el projecte al que es vol afegir el nou requisit. <i>(Extended from 2.3)</i>	
Req. coberts:	RF 2.3
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan es vol afegir un requisit a un projecte.	
	2. El sistema mostra el diàleg per escollir el tipus de requisit a afegir.
3. L'usuari introdueix el nom del requisit i selecciona que vol afegir un requisit de tipus "nou" i accepta.	
	4. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit de tipus nou.
5. L'usuari omple els camps i accepta.	
	6. El sistema afegeix el nou requisit al projecte.
Cursos Alternatius	
3a, 5a: el nom del requisit ja existeix dins del projecte	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
3b, 5b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
5c: el question text i/o el form text són buits o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació del requisit.

Cas d'ús #	2.3.2
Nom:	
Crear un requisit amb un patró associat	
Descripció:	
Es crea un requisit amb un patró associat i s'afegeix al projecte.	
Precondició:	
La llista dels patrons i dels seus elements està carregada. <i>(Extended from 2.3)</i> Existeix el projecte al que es vol afegir el nou requisit. <i>(Extended from 2.3)</i>	
Req. coberts:	RF 2.3
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan es vol afegir un requisit a un projecte.	
	2. El sistema mostra el diàleg per escollir el tipus de requisit a afegir.
3. L'usuari introdueix el nom del requisit i selecciona que vol afegir un requisit amb un patró associat.	
4. L'usuari selecciona el patró o el requirement form amb el que vol associar el requisit de la finestra amb els patrons.	
5. L'usuari accepta.	
	6. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit amb un patró associat.
7. L'usuari omple els camps i accepta.	
	8. El sistema afegeix el nou requisit al projecte.
Cursos Alternatius	
5a, 7a: el nom del requisit ja existeix dins del projecte	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
5b, 7b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
5c: l'usuari no selecciona cap patró ni requirement form per associar al requisit	
	1. El sistema mostra el missatge corresponent i es torna al diàleg per escollir el tipus de requisit.
7c: el question text i/o el form text són buits o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació del requisit.

Cas d'ús #	2.3.3
Nom:	
Crear un requisit instància d'un patró	
Descripció:	
Es crea un requisit instància d'un patró i s'afegeix al projecte.	
Precondició:	
La llista dels patrons i dels seus elements està carregada. <i>(Extended from 2.3)</i> Existeix el projecte al que es vol afegir el nou requisit. <i>(Extended from 2.3)</i>	
Req. coberts:	RF 2.3
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan es vol afegir un requisit a un projecte.	
	2. El sistema mostra el diàleg per escollir el tipus de requisit a afegir.
3. L'usuari introdueix el nom del requisit i selecciona que vol afegir un requisit instància d'un patró.	
4. L'usuari selecciona la fixed o extended part que vol instanciar de la finestra amb els patrons.	
5. L'usuari accepta.	
	6. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit instància d'un patró.
7. L'usuari omple els camps.	
8. Opcionalment, l'usuari pot modificar el valor d'un paràmetre (veure cas d'ús 2.6).	
9. L'usuari accepta.	
	10. El sistema afegeix el nou requisit al projecte.
Cursos Alternatius	
5a, 9a: el nom del requisit ja existeix dins del projecte	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
5b, 9b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg des d'on s'ha produït l'error.
5c: l'usuari no selecciona cap fixed o extended part per ser instanciada	
	1. El sistema mostra el missatge corresponent i es torna al diàleg per escollir el tipus de requisit.
7c: el question text i/o el form text són buits o un conjunt d'espais en blanc	

	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació del requisit.
9c: el form text no té tots els paràmetres que si té el fixed/extended part que el requisit instància	
	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació del requisit.
9d: algun valor no està definit	
	1. El sistema mostra el missatge corresponent i es torna al diàleg de creació del requisit.

El següent cas d'ús, el 2.4, té un comportament diferent i una interacció diferent segons el tipus de requisit que l'usuari vulgui editar. És per això que es tracten com a subcasos d'aquest:

- 2.4.1: Editar un requisit “nou”.
- 2.4.2: Editar un requisit amb un patró associat.
- 2.4.3: Editar un requisit instància d'un patró.

Cas d'ús #	2.4
Nom:	
Editar un requisit d'un projecte	
Descripció:	
S'actualitza dins del projecte el requisit que es vol editar amb els nous canvis.	
Precondició:	
La llista de patrons i dels seus elements està carregada. Existeix el projecte. El projecte conté el requisit que es vol editar.	
Req. coberts:	RF 2.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Actor
1. El cas d'ús comença quan l'usuari selecciona per editar un requisit d'un projecte.	
	<i>[vegis subapartat corresponent segons el tipus de requisit que es vol editar]</i>
Cursos Alternatius	

Cas d'ús #	2.4.1
Nom:	
Editar un requisit "nou"	
Descripció:	
S'actualitza dins del projecte el requisit que es vol editar amb els nous canvis.	
Precondició:	
La llista de patrons i dels seus elements està carregada. (<i>Extended from 2.4</i>)	
Existeix el projecte. (<i>Extended from 2.4</i>)	
El projecte conté el requisit que es vol editar. (<i>Extended from 2.4</i>)	
Req. coberts:	RF 2.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan l'usuari selecciona per editar un requisit de tipus "nou" en un projecte.	
	2. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit de tipus "nou".
3. L'usuari edita els camps que consideri oportú.	
4. Opcionalment, l'usuari pot assignar un patró associat a aquest requisit, seleccionant de la finestra de patrons un patró o form i associant-lo.	4a. El requisit passa a ser un requisit amb un patró associat i deixa de ser un requisit "nou".
5. L'usuari accepta.	
	6. El sistema actualitza dins del projecte el requisit amb els nous canvis.
Cursos Alternatius	
3a: el question text i/o el form text són buits o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
5a: el nom del requisit ja existeix dins del projecte	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
5b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.

Cas d'ús #	2.4.2
Nom:	
Editar un requisit amb un patró associat.	
Descripció:	
S'actualitza dins del projecte el requisit que es vol editar amb els nous canvis.	
Precondició:	
La llista de patrons i dels seus elements està carregada. <i>(Extended from 2.4)</i>	
Existeix el projecte. <i>(Extended from 2.4)</i>	
El projecte conté el requisit que es vol editar. <i>(Extended from 2.4)</i>	
Req. coberts:	RF 2.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan l'usuari selecciona per editar un requisit amb un patró associat en un projecte.	
	2. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit amb un patró associat.
3. L'usuari edita els camps que consideri oportú.	
4. Opcionalment, l'usuari pot modificar el patró associat a aquest requisit, seleccionant de la finestra de patrons un patró o form i associant-lo.	
5. Opcionalment, l'usuari pot modificar el tipus del requisit a un de tipus "nou" desassociant el patró que té associat.	5a. El requisit passa a ser de tipus "nou".
6. L'usuari accepta.	
	7. El sistema actualitza dins del projecte el requisit amb els nous canvis.
Cursos Alternatius	
3a: el question text i/o el form text són buits o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
6a: el nom del requisit ja existeix dins del projecte	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
6b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.

Cas d'ús #	2.4.3
Nom:	
Editar un requisit instància d'un patró.	
Descripció:	
S'actualitza dins del projecte el requisit que es vol editar amb els nous canvis.	
Precondició:	
La llista de patrons i dels seus elements està carregada. (<i>Extended from 2.4</i>)	
Existeix el projecte. (<i>Extended from 2.4</i>)	
El projecte conté el requisit que es vol editar. (<i>Extended from 2.4</i>)	
Req. coberts:	RF 2.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan l'usuari selecciona per editar un requisit instància d'un patró en un projecte.	
	2. El sistema mostra el diàleg des d'on es poden editar els atributs d'un requisit instància d'un patró.
3. L'usuari edita els camps que consideri oportú.	
4. Opcionalment, l'usuari pot modificar el patró que instància el requisit, seleccionant de la finestra de patrons una fixed o extended part i associant-lo.	4a. El sistema modifica al diàleg i al requisit el question text, el form text i els paràmetres amb els de la fixed/extended part que el requisit ara instància.
5. Opcionalment, l'usuari pot modificar seleccionant l'acció corresponent modificar el question text i el form text del requisit amb els del patró que instància.	5a. El sistema modifica al diàleg i al requisit el question text i el form text amb els del patró que el requisit instància.
6. Opcionalment, l'usuari pot modificar el valor d'un paràmetre (veure cas d'ús 2.6).	
7. Opcionalment, l'usuari pot modificar el tipus del requisit a un requisit amb patró associat seleccionant l'opció corresponent.	7a. El requisit passa a ser de tipus requisit amb patró associat i té com a patró associat el patró que instanciava.
8. L'usuari accepta.	
	9. El sistema actualitza dins del projecte el requisit amb els nous canvis.
Cursos Alternatius	
3a: el question text i/o el form text són buits o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
8a: el nom del requisit ja existeix dins del projecte	

	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
8b: el nom del requisit és buit o un conjunt d'espais en blanc	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
8c: el form text no té tots els paràmetres que si té el fixed/extended part que el requisit instància	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.
8d: algun valor no està definit	
	1. El sistema mostra el missatge corresponent i es torna al diàleg d'edició del requisit.

Cas d'ús #	2.5
Nom:	
Eliminar un requisit d'un projecte	
Descripció:	
S'elimina del projecte el requisit seleccionat.	
Precondició:	
Existeix el projecte. El projecte conté el requisit que es vol eliminar.	
Req. coberts:	RF 2.5
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona el requisit que vol suprimir del projecte i acciona la funció d'eliminar.	
	2. El sistema elimina el requisit i tots els seus valors, si és que en tenia algun.
Cursos Alternatius	

Cas d'ús #	2.6
Nom:	
Modificar el valor d'un requisit	
Descripció:	
S'actualitza dins del requisit el valor associat al paràmetre amb el nou valor.	
Precondició:	
Existeix el requisit. El requisit conté el valor que es vol modificar.	
Req. coberts:	RF 2.4
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. El cas d'ús comença quan l'usuari selecciona per editar el valor d'un paràmetre en un requisit.	
	2. El sistema mostra el diàleg per modificar el valor.
3. L'usuari modifica els atributs del valor que consideri oportú i accepta.	
	4. El sistema actualitza al requisit el valor amb els canvis realitzats.
Cursos Alternatius	
3a: el nou valor no satisfà les restriccions del paràmetre que té associat.	
	1. El sistema mostra el missatge corresponent i pregunta a l'usuari si està segur de que vol afegir un valor que no satisfà les restriccions.
2. L'usuari pot acceptar o cancel·lar.	
	3a. Si l'usuari accepta: El sistema actualitza al requisit el valor amb els canvis realitzats.
	3b. Si l'usuari cancel·la: Es torna al diàleg d'edició del valor.

12.3.3. Gestió de projectes

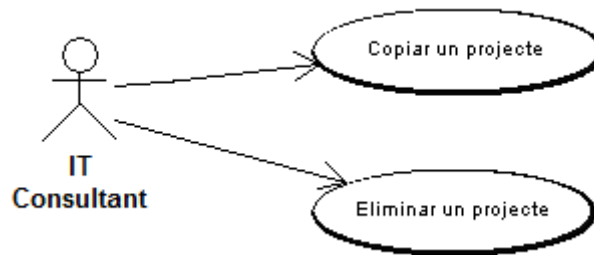


Figura 17: Diagrama de casos d'ús sobre *Gestió de projectes*

Cas d'ús #	3.1
Nom:	
Copiar un projecte	
Descripció:	
Es copia el projecte i tots els seus requisits en un nou projecte amb un nom diferent.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte a copiar.	
Req. coberts:	RF 3.1
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona el projecte que vol copiar i acciona la funció de copiar.	
	2. El sistema copia el projecte i tot el seu contingut en un nou projecte amb un nom diferent que cap altre projecte tingui.
Cursos Alternatius	

Cas d'ús #	3.2
Nom:	
Eliminar un projecte	
Descripció:	
S'elimina el projecte seleccionat.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte que es vol eliminar.	
Req. coberts:	RF 3.2
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona el projecte que vol suprimir i acciona la funció d'eliminar.	
	2. El sistema elimina el projecte i tot el seu contingut.
Cursos Alternatius	

12.3.4. Interoperabilitat

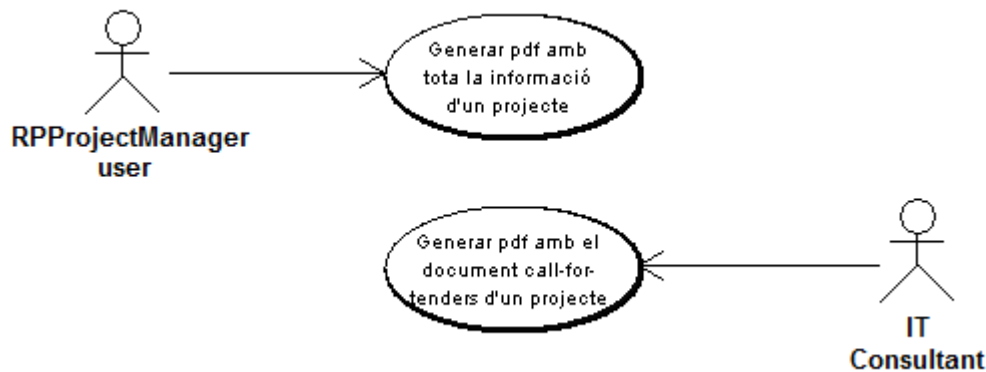


Figura 18: Diagrama de casos d'ús sobre *Interoperabilitat*

Cas d'ús #	4.1
Nom:	
Generar pdf amb tota la informació d'un projecte	
Descripció:	
Es genera un document pdf amb tota la informació del projecte seleccionat.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte sobre el que es genera el pdf.	
Req. coberts:	RF 4.1
Actors:	IT Consultant / Requirements Expert
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un projecte i acciona la funció de generar un pdf amb tota la informació.	
	2. El sistema genera el pdf i mostra un diàleg per seleccionar on es desitja salvar el fitxer.
3. L'usuari selecciona el fitxer en el que es guardarà el pdf.	
	4. El sistema emmagatzema a la ruta seleccionada per l'usuari el document pdf que ha generat.
Cursos Alternatius	
3a: el fitxer seleccionat ja existeix.	
	1. El sistema pregunta a l'usuari si està segur de que vol sobreescriure el fitxer.
2. L'usuari pot acceptar o cancel·lar.	

	3a. Si l'usuari accepta: El sistema sobreescriu el fitxer amb el pdf generat.
	3b. Si l'usuari cancel·la: Es torna al diàleg de selecció del fitxer.
3b: no es pot escriure al fitxer seleccionat perquè una altra aplicació el té obert.	
	1. El sistema mostra el missatge corresponent i acaba.

Cas d'ús #	4.2
Nom:	
Generar pdf amb el document call-for-tenders d'un projecte.	
Descripció:	
Es genera un document pdf amb el document call-for-tenders del projecte seleccionat.	
Precondició:	
La llista de projectes està carregada. Existeix el projecte sobre el que es genera el pdf.	
Req. coberts:	RF 4.2
Actors:	IT Consultant
Curs típic d'esdeveniments	
Actor	Sistema
1. L'usuari selecciona un projecte i acciona la funció de generar un pdf amb el document call-for-tenders.	
	2. El sistema genera el pdf i mostra un diàleg per seleccionar on es desitja salvar el fitxer.
3. L'usuari selecciona el fitxer en el que es guardarà el pdf.	
	4. El sistema emmagatzema a la ruta seleccionada per l'usuari el document pdf que ha generat.
Cursos Alternatius	
3a: el fitxer seleccionat ja existeix.	
	1. El sistema pregunta a l'usuari si està segur de que vol sobreescriure el fitxer.
2. L'usuari pot acceptar o cancel·lar.	
	3a. Si l'usuari accepta: El sistema sobreescriu el fitxer amb el pdf generat.
	3b. Si l'usuari cancel·la: Es torna al diàleg de selecció del fitxer.
3b: no es pot escriure al fitxer seleccionat perquè una altra aplicació el té obert.	
	1. El sistema mostra el missatge corresponent i acaba.

13. Model Conceptual

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79	
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
13.1. Diagrama de classes.....	126
13.2. Restriccions d'integritat textuals.....	129
13.3. Matriu de traçabilitat	130
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

13.1. Diagrama de classes

A les següents Figures 19 i 20 es mostra el diagrama de classes de l'aplicació *RPPProjectManager*; per a que sigui més fàcil de seguir, s'ha separat en dues parts: la de projectes i la de valors. En aquest model podem trobar diferents tipus d'objectes, que podem dividir en diferents categories:

- *Project*. Com el seu nom indica, representa a un projecte. Un projecte pot tenir varis *Requirement* associats.
- *Requirement*. Representa als requisits. Els requisits poden ser de diferents tipus: o bé de tipus *PatternInstance* (quan instancien una fixed/extended part d'un patró) o bé de tipus *NoPatternInstance* (en qualsevol altre cas). Un requisit només pot pertànyer a un projecte.
- *NoPatternInstance*. Són aquells requisits que no instancien cap patró. Els requisits de tipus *NoPatternInstance* poden ser o no de tipus *WithPatternAssociated*. Els requisits que són de tipus *NoPatternInstance* però no són de tipus *WithPatternAssociated* són aquells requisits que s'han creat completament des de zero per falta de que existís un patró que s'adaptés a les necessitats que el requisit ha de cobrir.
- *WithPatternAssociated*. Representen a aquells requisits que tenen relació amb un patró, però que per manca de que el patró contingui algun form o extended part que s'adapti a les necessitats que el requisit ha de cobrir no s'ha acabat podent instanciar. Aquest tipus de requisits tenen relació o bé amb un *RequirementPattern* (quan no existeix cap form que s'adapti) o bé amb un *RequirementForm* (quan no existeix cap fixed/extended part que s'adapti). En aquest últim cas, el patró que té associat el requisit és el *RequirementPattern* al que pertany el *RequirementForm* que té associat.
- *PatternInstance*. Representa als requisits que instancien un *PatternItem* (una fixed o extended part) d'un patró. Per tant, un requisit d'aquest tipus té associat el *PatternItem* que instància, així com també el *RequirementForm* i el *RequirementPattern* al que pertany el *PatternItem*. A més, poden tenir associats zero o més *Value* (tants com paràmetres tingui el *PatternItem* que instància).
- *Value*. Representa el valor que pot prendre un paràmetre dins d'un requisit que instància un patró. Els valors poden ser de tipus *SimpleValue* (un únic valor) o *SetValue* (un conjunt de valors simples). Els valors simples poden ser de tipus *IntegerValue*, *FloatValue*, *StringValue*, *TimePointValue* (un instant) o *DomainValue* (un valor d'una enumeració de valors possibles). Els valors simples associats a un valor de tipus set no tenen associat cap paràmetre ni estan relacionats directament amb un *PatternInstance*, ja que és el seu set qui té aquestes relacions. Un valor simple satisfà les restriccions quan s'ajusta a les condicions imposades per la mètrica del paràmetre que el valor o el set al que pertany té associat (en el cas dels enters, per exemple, un valor de 50 en una mètrica amb mínim 10 i màxim 60 satisfà la restricció, però si el valor fos 100 no satisfaria la restricció).

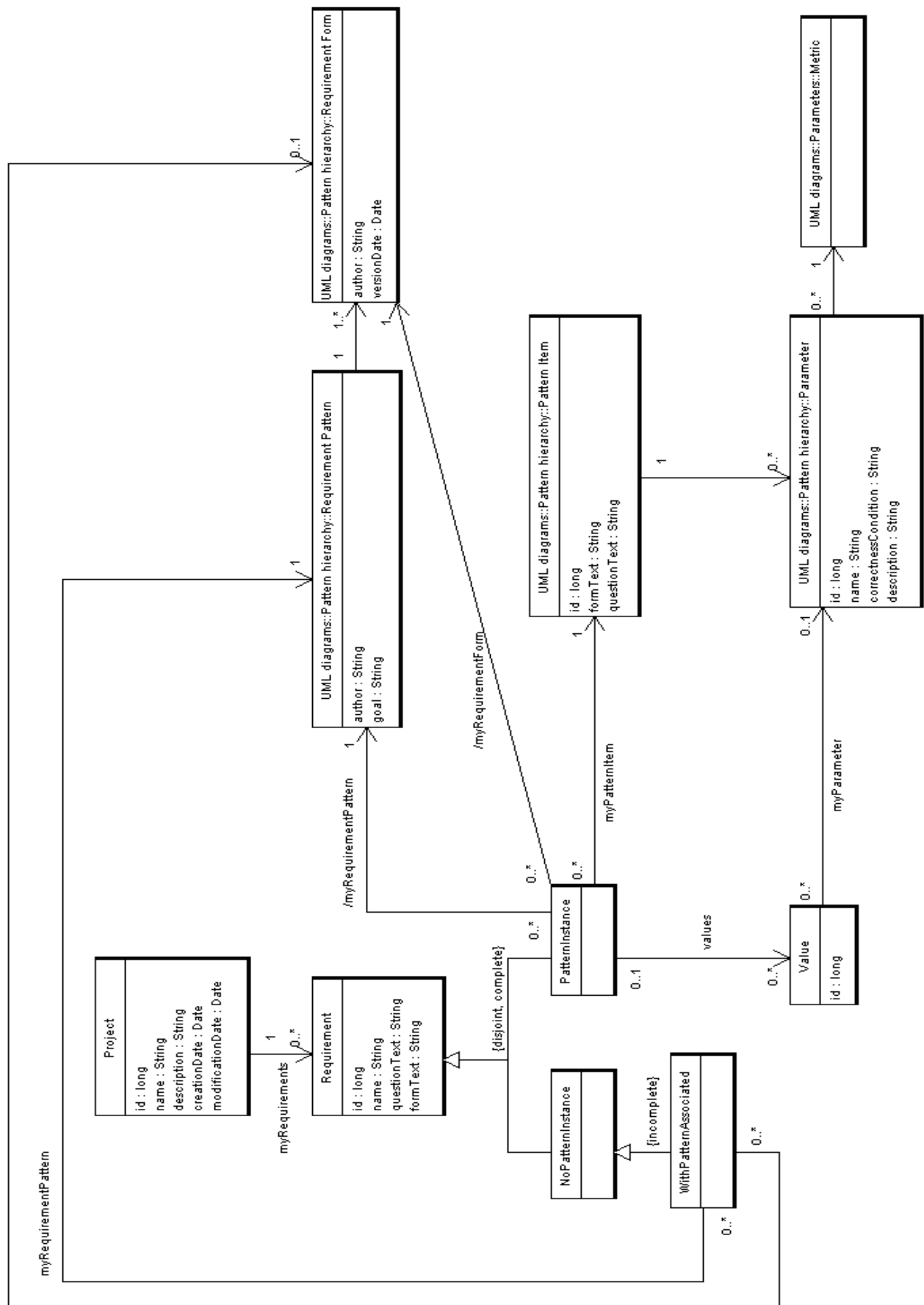


Figura 19: Diagrama de classes, part de projectes

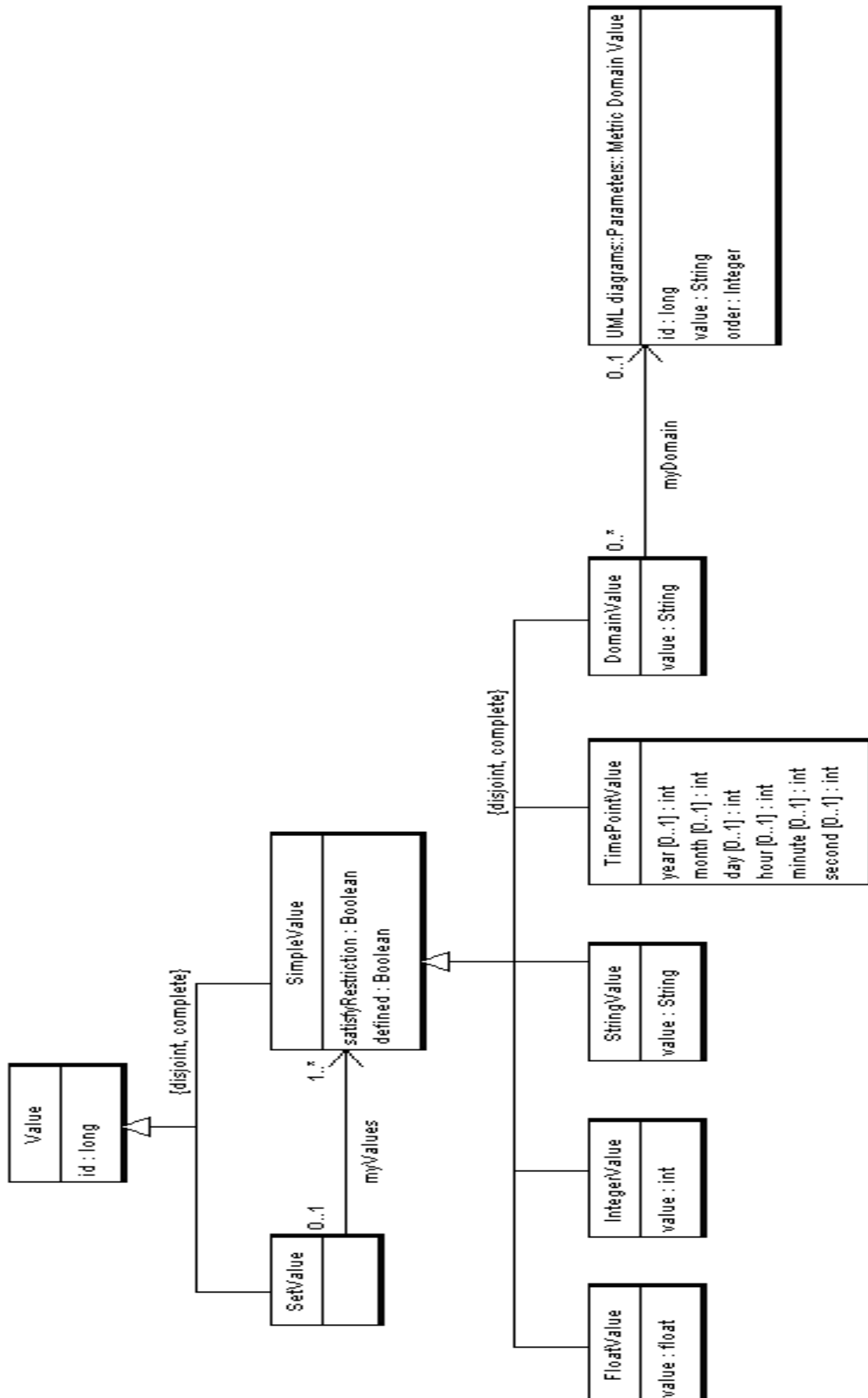


Figura 20: Diagrama de classes, part de valors

13.2. Restriccions d'integritat textuais

Aquestes són les restriccions d'integritat textuais necessàries per representar de manera precisa el domini i que no han pogut ser expressades en els diagrames anteriors:

- RI-1.** No pot haver-hi dos *Project* amb el mateix *id*.
- RI-2.** No pot haver-hi dos *Project* amb el mateix *name*.
- RI-3.** No pot haver-hi dos *Requirement* amb el mateix *id*.
- RI-4.** No pot haver-hi dos *Requirement* amb el mateix *name* dins del mateix projecte.
- RI-5.** Si un requisit de tipus *WithPatternAssociated* té un *RequirementForm* associat, el *RequirementPattern* associat ha de ser el patró al que pertany el *RequirementForm*.
- RI-6.** En un *PatternInstance*, el *RequirementForm* associat és el form al que pertany el *PatternItem* que té associat.
- RI-7.** En un *PatternInstance*, el *RequirementPattern* associat és el patró al que pertany el *RequirementForm* que té associat.
- RI-8.** Un *PatternInstance* ha de tenir tants *Value* com *Parameter* tingui el *PatternItem* que té associat i per cada *Parameter* d'aquest *PatternItem* el *PatternInstance* ha de tenir un *Value* que el tingui associat.
- RI-9.** No pot haver-hi dos *Value* amb el mateix *id*.
- RI-10.** Un *SimpleValue* o bé té un *Parameter* i *PatternInstance* associats i no pertany a un *SetValue* o bé no té cap *Parameter* ni *PatternInstance* associat però pertany a un *SetValue*.
- RI-11.** Un *SetValue* sempre té un *Parameter* i un *PatternInstance* associat.
- RI-12.** En un *SetValue*, tots els *SimpleValue* han de ser del mateix tipus.
- RI-13.** Un *Value* ha de ser:
 1. Si té un *Parameter* associat, del mateix tipus que la mètrica d'aquest *Parameter*.
 2. Si no té un *Parameter* associat, i per tant pertany a un *SetValue*, del mateix tipus que la mètrica simple associada al paràmetre del seu *SetValue*.
- RI-14.** Si un *DomainValue* té un *MetricDomainValue* associat, aquest ha de ser:
 1. En cas que tingui un *Parameter* associat, un dels valors possibles de la mètrica del *Parameter* associat.

2. En cas que el valor pertanyi a un *SetValue*, un dels valors possibles de la mètrica simple de la mètrica tipus set del *Parameter* que el *SetValue* al que pertany té associat.

- RI-15.** Si un *DomainValue* té un *MetricDomainValue* associat, el *value* del *DomainValue* ha de ser el mateix que el *value* del *MetricDomainValue* associat.
- RI-16.** L'atribut *name* d'un *Project* o *Requirement* no pot ser un string buit ni un conjunt d'espais en blanc.
- RI-17.** L'atribut *value* d'un *StringValue* o *DomainValue* no pot ser un string buit ni un conjunt d'espais en blanc.
- RI-18.** L'atribut *formText* i *questionText* d'un *Requirement* no pot ser un string buit ni un conjunt d'espais en blanc.
- RI-19.** L'atribut *formText* d'un *PatternInstance* ha de contenir el nom de tots els paràmetres del *PatternItem* que té associat.

13.3. Matriu de traçabilitat

A continuació presentem la matriu de traçabilitat de l'aplicació *RPPProjectManager*, que conté per cada cas d'ús definit (veure capítol anterior 12) les classes del model conceptual que es poden donar d'alta (A), actualitzar (U), eliminar (D) o consultar (C) mitjançant el cas d'ús en qüestió.

Classe	Cas d'ús									
	1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3.1	2.3.2
Project	C	C	C				A	A		
Requirement			C						A	A
NoPattern Instance			C						A	
WithPattern Associated			C							A
Pattern Instance			C							
Value			C							
SetValue			C							
SimpleValue			C							
FloatValue			C							
IntegerValue			C							
StringValue			C							
TimePointValue			C							
DomainValue			C							
Requirement Pattern			C	C	C	C			C	C

Requirement Form			C			C			C	C
PatternItem			C			C				
Parameter			C			C				
Metric						C				
MetricDomain Value						C				

Taula 6: Matriu de traçabilitat de RPPProjectManager, casos d'ús 1.1 a 2.3.2

Classe	Cas d'ús									
	2.3.3	2.4.1	2.4.2	2.4.3	2.5	2.6	3.1	3.2	4.1	4.2
Project							A,C	D	C	C
Requirement	A	U	U	U	D		A,C	D	C	C
NoPattern Instance		U			D		A,C	D		C
WithPattern Associated			U		D		A,C	D		C
Pattern Instance	A			U	D		A,C	D		C
Value				A,D	D	U	A,C	D		C
SetValue				A,D	D	U	A,C	D		C
SimpleValue				A,D	D	U	A,C	D		C
FloatValue				A,D	D	U	A,C	D		C
IntegerValue				A,D	D	U	A,C	D		C
StringValue				A,D	D	U	A,C	D		C
TimePointValue				A,D	D	U	A,C	D		C
DomainValue				A,D	D	U	A,C	D		C
Requirement Pattern	C	C	C	C			C			C
Requirement Form	C	C	C	C			C			C
PatternItem	C			C			C			C
Parameter	C			C		C	C			C
Metric						C				
MetricDomain Value						C	C			

Taula 7: Matriu de traçabilitat de RPPProjectManager, casos d'ús 2.3.3 a 4.2

14. Model de Comportament

1.	Introducció	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades.....	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
14.1.	<i>Introducció</i>	<i>134</i>
14.2.	<i>Visualització de projectes i patrons.....</i>	<i>135</i>
14.3.	<i>Construcció de projectes</i>	<i>141</i>
14.4.	<i>Gestió de projectes</i>	<i>162</i>
14.5.	<i>Interoperabilitat</i>	<i>164</i>
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

14.1. Introducció

En aquest capítol es mostra el model de comportament del sistema. En aquest sentit, per cada cas d'ús a implementar, es mostren els diagrames de seqüència del sistema (seqüència d'esdeveniments entre els actors i el sistema amb els que es poden identificar les operacions del sistema).

Per cada operació del sistema es descriu també el contracte de l'operació. Aquest contracte es descriu utilitzant la següent fitxa:

Nom Operació
Context:
Nom de l'operació amb els seus paràmetres i sortides
Precondició:
Precondició de l'operació
Excepcions:
Excepcions possibles i la resposta del sistema en aquests casos
Sortida:
Resultat de retorn de l'operació
Postcondició:
Postcondició de l'operació

14.2. Visualització de projectes i patrons

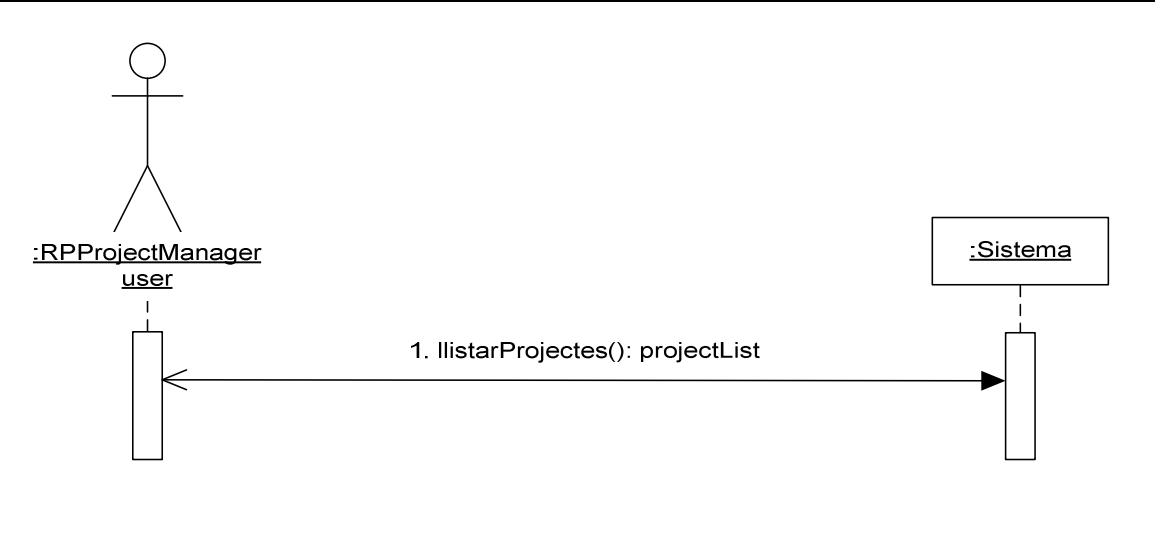
Visualitzar tots els projectes en ordre alfabètic

Casos d'ús coberts:

Use Case 1.1

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

llistarProjectes

Context:

llistarProjectes(): projectList

Precondició:

Excepcions:

Sortida:

La llista dels projectes que hi ha al repositori de l'aplicació.

Postcondició:

Es mostra la llista de tots els projectes en ordre alfabètic.

Visualitzar els atributs d'un projecte

Casos d'ús coberts:

Use Case 1.2

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectProjectFromList

Context:

selectProjectFromList(): project

Precondició:

La llista de projectes està carregada.

Excepcions:

Sortida:

El projecte seleccionat.

Postcondició:

Es mostren els atributs del projecte seleccionat.

Visualitzar tota la informació d'un projecte

Casos d'ús coberts:

Use Case 1.3

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectProjectFromList

Context:

selectProjectFromList(): project

Precondició:

La llista de projectes està carregada.

Excepcions:

Sortida:

El projecte seleccionat.

Postcondició:

Es mostra tota la informació del projecte seleccionat.

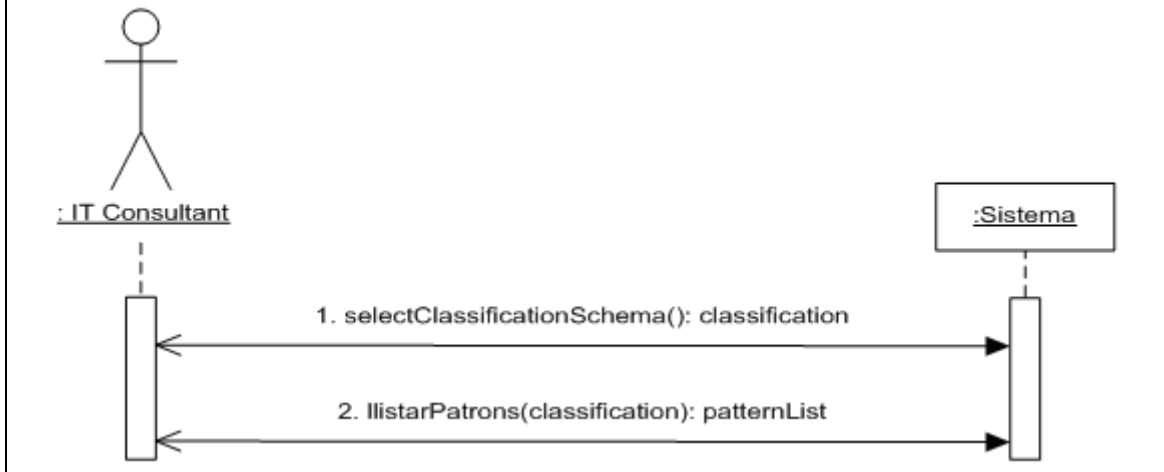
Visualitzar tots els patrons segons el classification schema escollit

Casos d'ús coberts:

Use Case 1.4

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectClassificationSchema

Context:

selectClassificationSchema(): classification

Precondició:

La llista dels classification schema està carregada.

Excepcions:

Sortida:

El classification schema seleccionat.

Postcondició:

llistarPatrons

Context:

llistarPatrons(classification): patternList

Precondició:

Excepcions:

Sortida:

La llista de patrons que hi ha al repositori de l'aplicació.

Postcondició:

Es mostra la llista de tots els patrons i dels seus elements (forms, extended i fixed parts) ordenada segons el classification schema passat com a paràmetre.

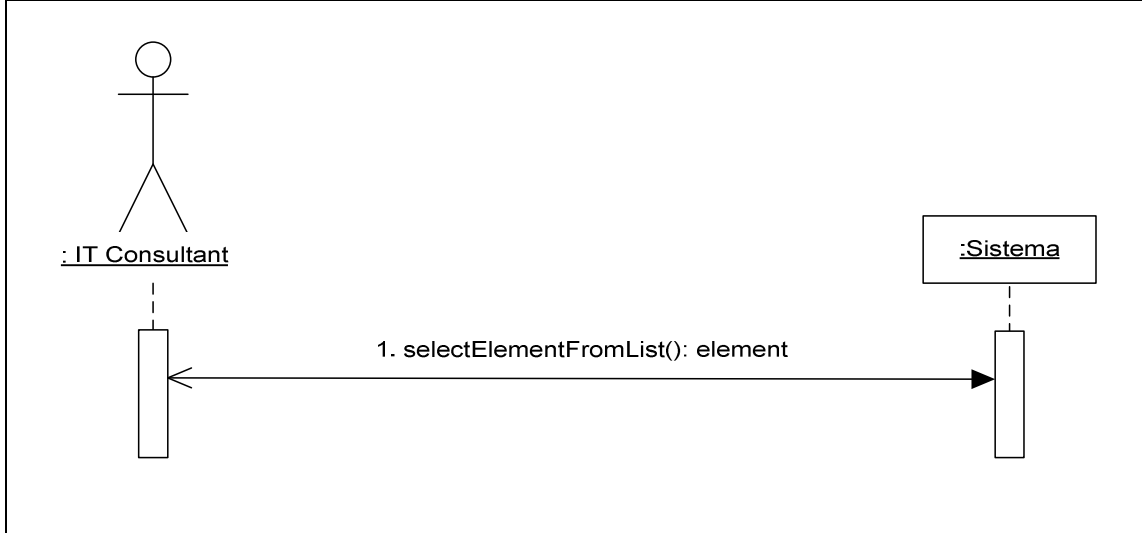
Visualitzar els atributs d'un element d'un patró

Casos d'ús coberts:

Use Case 1.5

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectElementFromList

Context:

selectElementFromList(): element

Precondició:

La llista de patrons i dels seus elements està carregada.

Excepcions:

Sortida:

L'element seleccionat.

Postcondició:

Es mostren els atributs de l'element seleccionat.

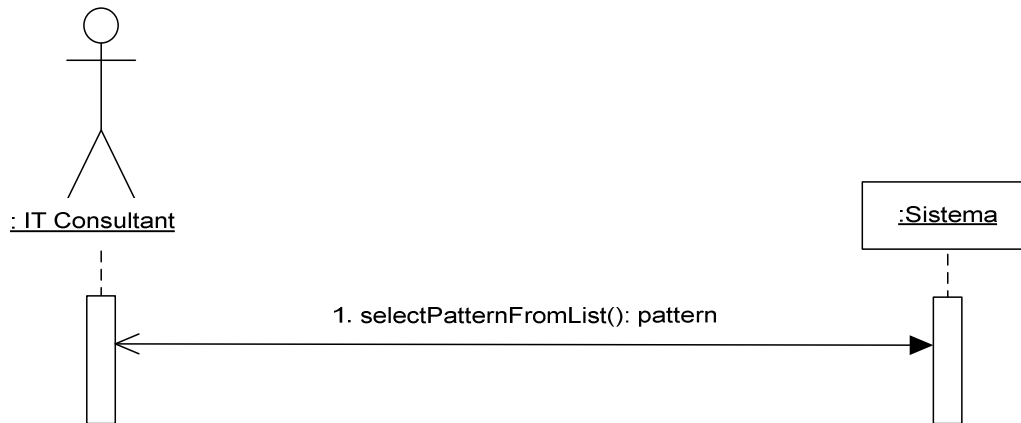
Visualitzar tota la informació d'un patró

Casos d'ús coberts:

Use Case 1.6

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectPatternFromList

Context:

selectPatternFromList(): pattern

Precondició:

La llista de patrons i dels seus elements està carregada.

Excepcions:

Sortida:

El patró seleccionat.

Postcondició:

Es mostra tota la informació del patró seleccionat.

14.3. Construcció de projectes

Crear un projecte

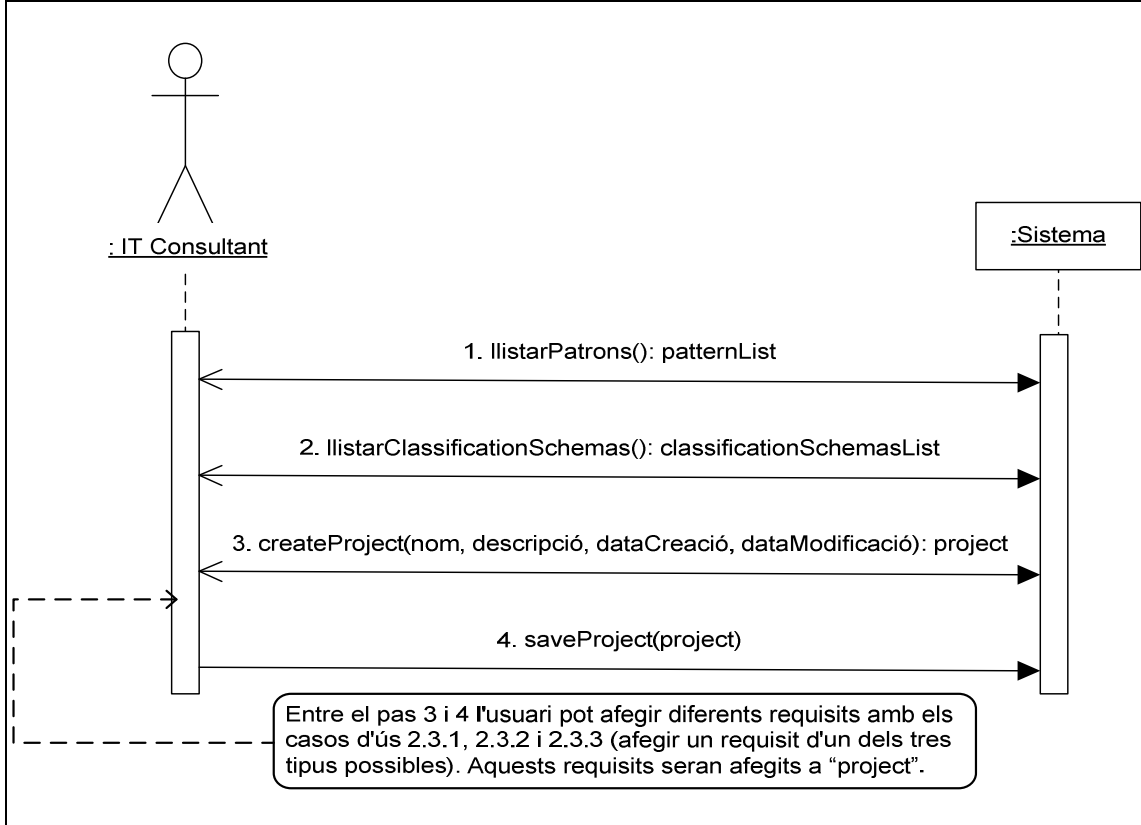
Casos d'ús coberts:

Use Case 2.1

Restriccions d'integritat a tenir en compte:

RI-1, RI-2, RI-16

Diagrama de seqüència



Contracte d'operacions

l·listarPatrons

Context:

l·listarPatrons(): patternList

Precondició:

Excepcions:

Sortida:

La llista de patrons que hi ha al repositori de l'aplicació.

Postcondició:

Es carrega i es mostra la llista de tots els patrons i dels seus elements (forms, extended i fixed parts) ordenada alfabèticament.

<i>listarClassificationSchemas</i>
Context:
listarClassificationSchemas(): classificationSchemasList
Precondició:

Excepcions:

Sortida:
La llista de tots els classification schema que hi ha al repositori de l'aplicació.
Postcondició:
Es carrega la llista de tots els classification schema ordenada alfabèticament.

<i>createProject</i>
Context:
createProject(nom, descripció, dataCreació, dataModificació): project
Precondició:

Excepcions:

Sortida:
El projecte creat.
Postcondició:

<i>saveProject</i>
Context:
saveProject(project)
Precondició:
No existeix cap projecte amb aquest nom al repositori ni és un conjunt d'espais en blanc.
Excepcions:
Si ja existeix un projecte amb aquest nom al repositori o el nom és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:

Postcondició:
El projecte rebut com a paràmetre junt amb tots els seus requisits queden desats al repositori.

Editar un projecte

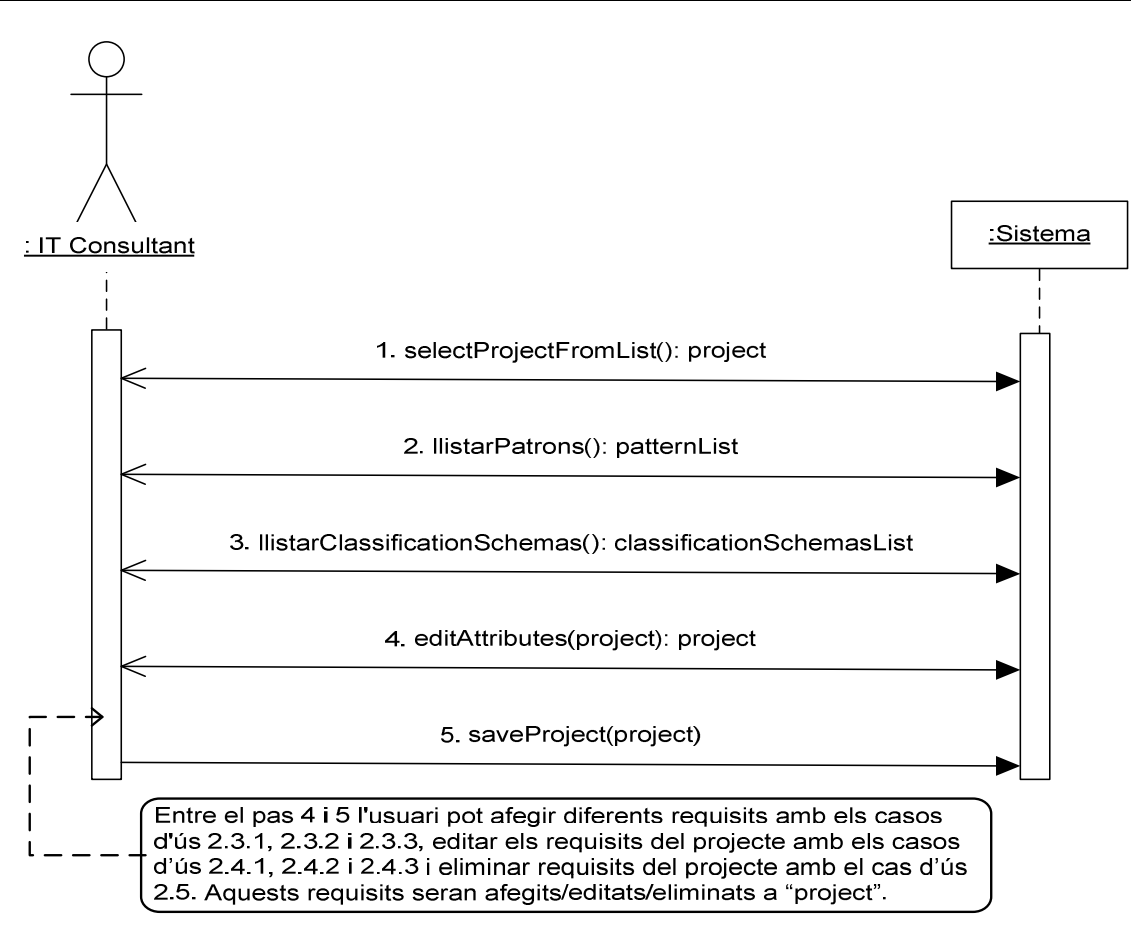
Casos d'ús coberts:

Use Case 2.2

Restriccions d'integritat a tenir en compte:

RI-1, RI-2, RI-16

Diagrama de seqüència



Contracte d'operacions

l·listarPatrons

Context:

l·listarPatrons(): patternList

Precondició:

Excepcions:

Sortida:

La llista de patrons que hi ha al repositori de l'aplicació.

Postcondició:

Es carrega i es mostra la llista de tots els patrons i dels seus elements (forms, extended i fixed parts) ordenada alfabèticament.

<i>selectProjectFromList</i>
Context:
selectProjectFromList(): project
Precondició:
La llista de projectes està carregada.
Excepcions:

Sortida:
El projecte seleccionat.
Postcondició:

<i>listarClassificationSchemas</i>
Context:
listarClassificationSchemas(): classificationSchemasList
Precondició:

Excepcions:

Sortida:
La llista de tots els classification schema que hi ha al repositori de l'aplicació.
Postcondició:
Es carrega la llista de tots els classification schema ordenada alfabèticament.

<i>editAttributes</i>
Context:
editAttributes(project): project
Precondició:

Excepcions:

Sortida:
El projecte amb els seus atributs actualitzats.
Postcondició:

<i>saveProject</i>
Context:
saveProject(project)
Precondició:
No existeix cap projecte amb aquest nom ni és un conjunt d'espais en blanc.
Excepcions:
Si ja existeix un projecte amb aquest nom al repositori o el nom és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:

Postcondició:
El projecte rebut com a paràmetre junt amb tots els seus requisits queden actualitzats al repositori.

Crear un requisit “nou”

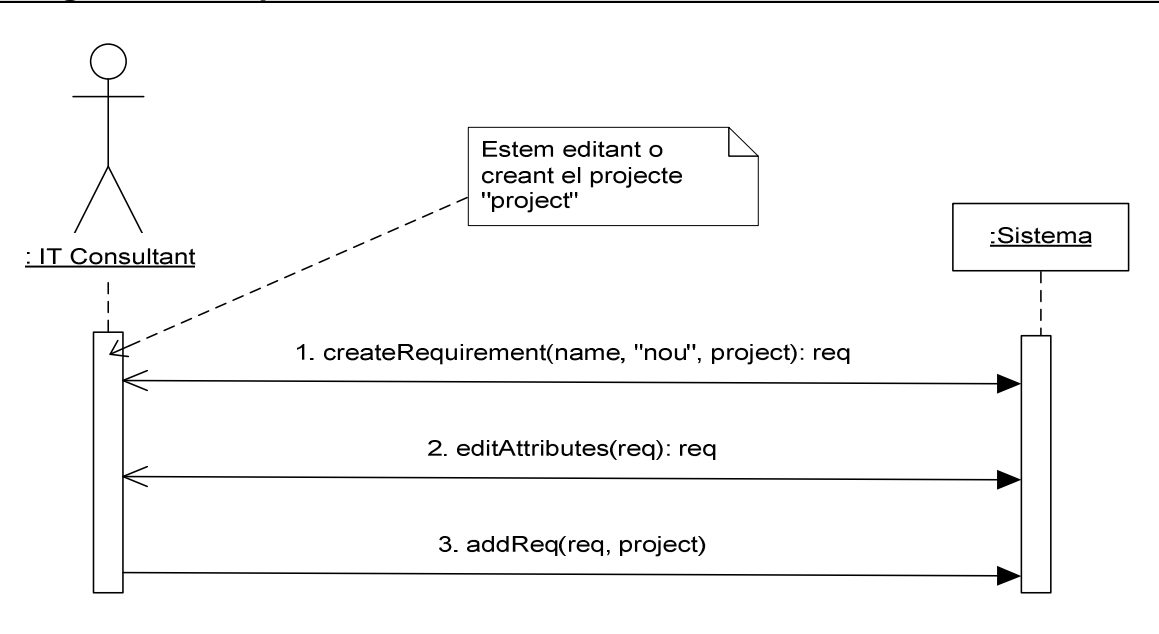
Casos d'ús coberts:

Use Case 2.3.1

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18

Diagrama de seqüència



Contracte d'operacions

createRequirement

Context:

`createRequirement(name, "nou", project): req`

Precondició:

No existeix cap requisit amb aquest nom a "project" ni és un conjunt d'espais en blanc.

Excepcions:

Si ja existeix un requisit amb aquest nom a "project" o el nom és un conjunt d'espais en blanc, el sistema informa de l'error.

Sortida:

El requisit de tipus "nou" creat.

Postcondició:

<i>editAttributes</i>	
Context:	
editAttributes(req): req	
Precondició:	

Excepcions:	

Sortida:	
El requisit amb els seus atributs actualitzats.	
Postcondició:	

<i>addReq</i>	
Context:	
addReq(req, project)	
Precondició:	
No existeix cap requisit amb el mateix nom a “project” ni el nom és un conjunt d’espais en blanc. Ni el question text ni el form text del requisit és un conjunt d’espais en blanc.	
Excepcions:	
Si ja existeix un requisit amb el mateix nom a “project” o el nom és un conjunt d’espais en blanc, el sistema informa de l’error. Si el question text o el form text del requisit és un conjunt d’espais en blanc, el sistema informa de l’error.	
Sortida:	

Postcondició:	
S’afegeix al projecte el requisit rebut com a paràmetre.	

Crear un requisit amb un patró associat

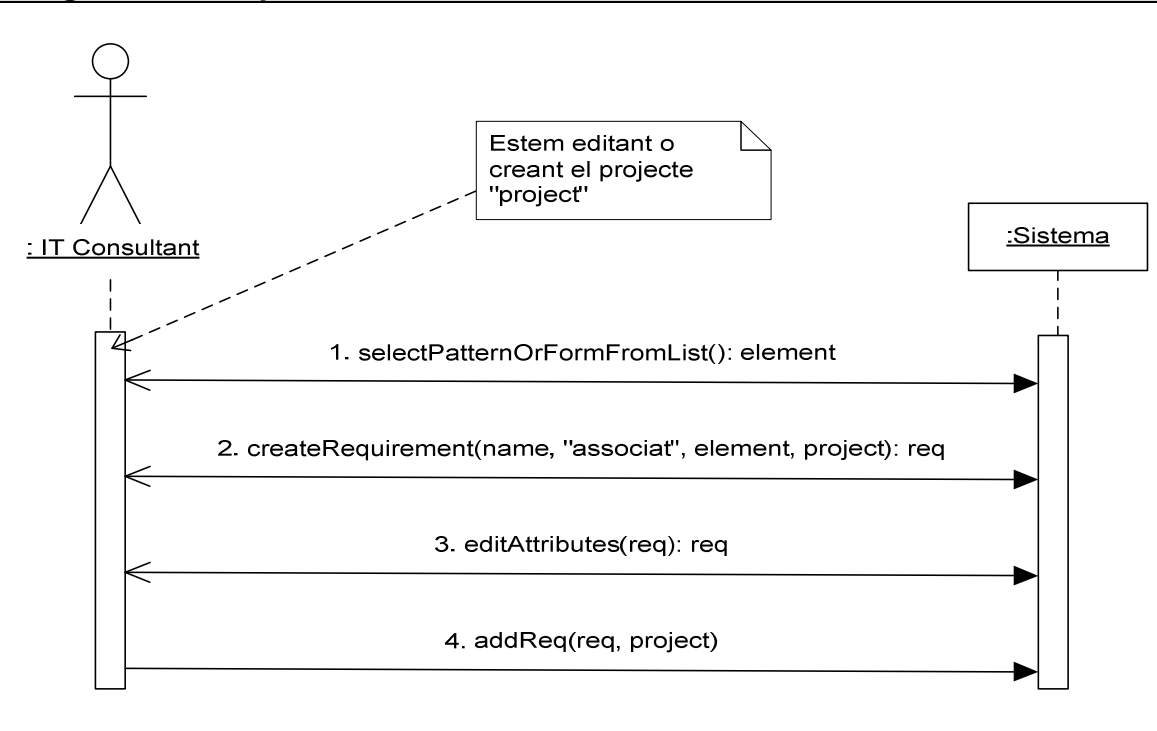
Casos d'ús coberts:

Use Case 2.3.2

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18, RI-5

Diagrama de seqüència



Contracte d'operacions

selectPatternOrFormFromList

Context:

selectPatternOrFormFromList(): element

Precondició:

La llista de patrons i dels seus elements està carregada.

Excepcions:

Sortida:

El patró o el requirement form seleccionat.

Postcondició:

<i>createRequirement</i>	
Context:	
createRequirement(name, "associat", element, project): req	
Precondició:	
No existeix cap requisit amb aquest nom a "project" ni és un conjunt d'espais en blanc. "element" ha de ser un patró o un requirement form.	
Excepcions:	
Si ja existeix un requisit amb aquest nom a "project" o el nom és un conjunt d'espais en blanc, el sistema informa de l'error. Si "element" no és un patró ni un requirement form, el sistema informa de l'error.	
Sortida:	
El requisit de tipus "requisit amb patró associat" creat.	
Postcondició:	

<i>editAttributes</i>	
Context:	
editAttributes(req): req	
Precondició:	

Excepcions:	

Sortida:	
El requisit amb els seus atributs actualitzats.	
Postcondició:	

<i>addReq</i>	
Context:	
addReq(req, project)	
Precondició:	
No existeix cap requisit amb el mateix nom a "project" ni el nom és un conjunt d'espais en blanc. Ni el question text ni el form text del requisit és un conjunt d'espais en blanc.	
Excepcions:	
Si ja existeix un requisit amb el mateix nom a "project" o el nom és un conjunt d'espais en blanc, el sistema informa de l'error. Si el question text o el form text del requisit és un conjunt d'espais en blanc, el sistema informa de l'error.	
Sortida:	

Postcondició:	
S'afegeix al projecte el requisit rebut com a paràmetre.	

Crear un requisit instància d'un patró

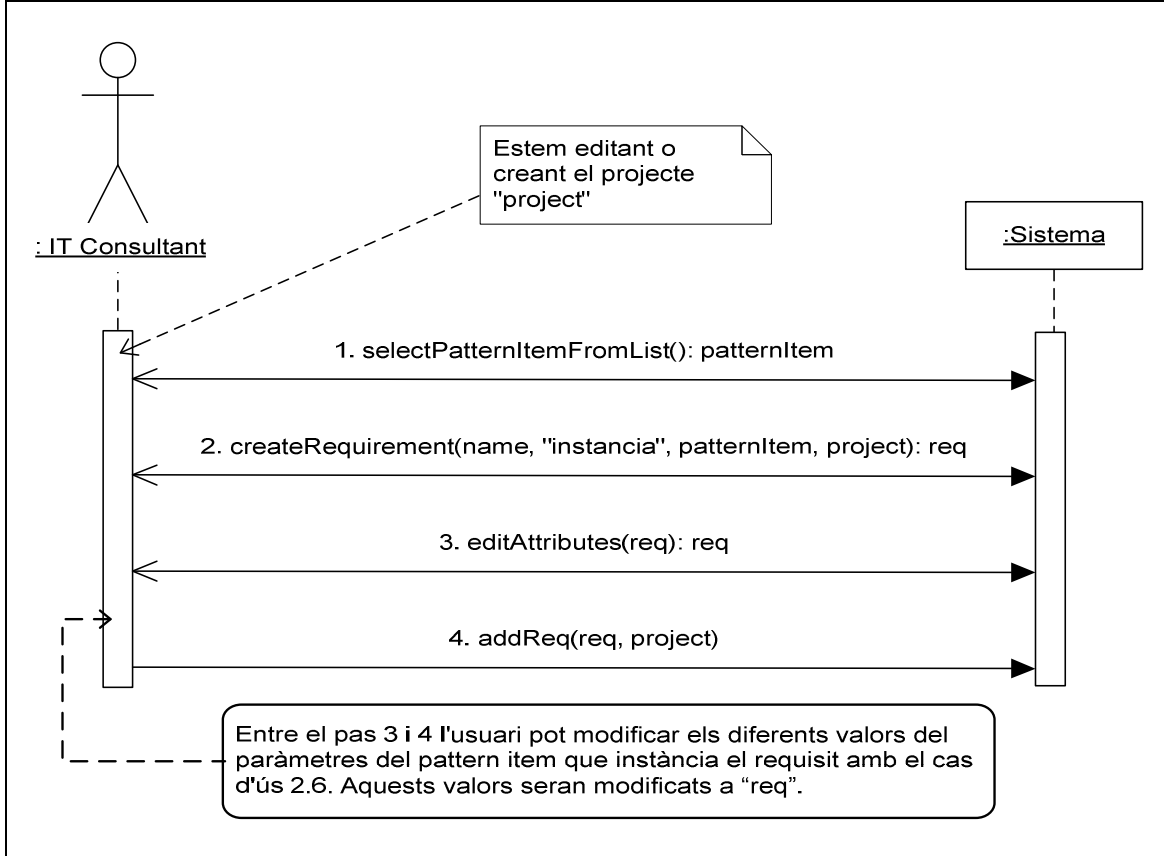
Casos d'ús coberts:

Use Case 2.3.3

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18, RI-6, RI-7, RI-8, RI-19

Diagrama de seqüència



Contracte d'operacions

selectPatternItemFromList

Context:

selectPatternItemFromList(): patternItem

Precondició:

La llista de patrons i dels seus elements està carregada.

Excepcions:

Sortida:

El pattern item seleccionat.

Postcondició:

<i>createRequirement</i>
Context:
createRequirement(name, "instancia", patternItem, project): req
Precondició:
No existeix cap requisit amb aquest nom a "project" ni és un conjunt d'espais en blanc. "element" ha de ser un pattern item.
Excepcions:
Si ja existeix un requisit amb aquest nom a "project" o el nom és un conjunt d'espais en blanc o si "element" no és un pattern item, el sistema informa de l'error.
Sortida:
El requisit de tipus "requisit instància d'un patró" creat.
Postcondició:

<i>editAttributes</i>
Context:
editAttributes(req): req
Precondició:

Excepcions:

Sortida:
El requisit amb els seus atributs actualitzats.
Postcondició:

<i>addReq</i>
Context:
addReq(req, project)
Precondició:
No existeix cap requisit amb el mateix nom a "project" ni el nom és un conjunt d'espais en blanc. Ni el question text ni el form text del requisit és un conjunt d'espais en blanc. El form text té tots els noms dels paràmetres del pattern item que el requisit instància i tots els valors del requisit estan definits.
Excepcions:
Si ja existeix un requisit amb el mateix nom a "project" o el nom és un conjunt d'espais en blanc, o si el question text o el form text del requisit és un conjunt d'espais en blanc, o si el form text no té el nom de tots els paràmetres del pattern item que el requisit instància, o si algun valor del requisit no està definit, el sistema informa de l'error.
Sortida:

Postcondició:
S'afegeix al projecte el requisit rebut com a paràmetre.

Editar un requisit “nou”

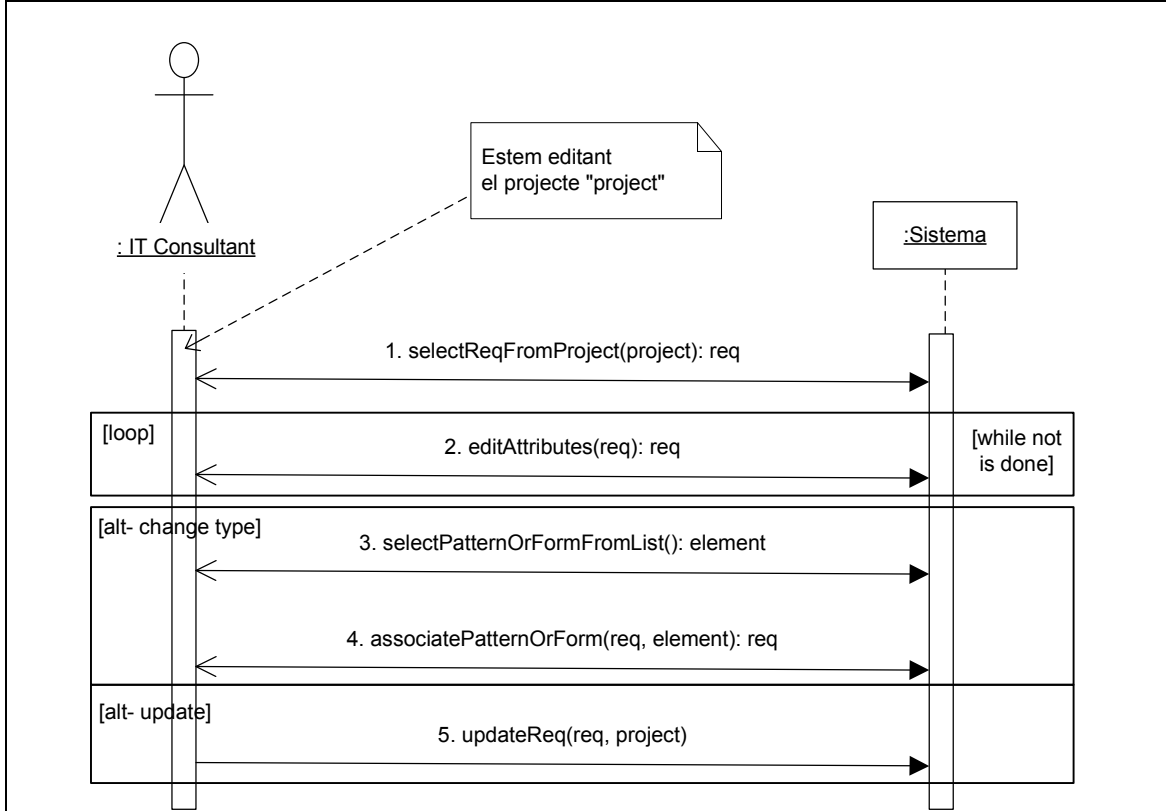
Casos d'ús coberts:

Use Case 2.4.1

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18, RI-5

Diagrama de seqüència



Contracte d'operacions

selectReqFromProject

Context:

selectReqFromProject(project): req

Precondició:

Excepcions:

Sortida:

El requisit de tipus “nou” seleccionat.

Postcondició:

editAttributes

Context:

editAttributes(req): req

Precondició:

Ni el question text ni el form text del requisit és un conjunt d'espais en blanc.

Excepcions:
Si el question text o el form text del requisit és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:
El requisit amb els seus atributs actualitzats.
Postcondició:

<i>selectPatternOrFormFromList</i>
Context:
selectPatternOrFormFromList(): element
Precondició:
La llista de patrons i dels seus elements està carregada.
Excepcions:

Sortida:
El patró o el requirement form seleccionat.
Postcondició:

<i>associatePatternOrForm</i>
Context:
associatePatternOrForm(req, element): req
Precondició:

Excepcions:

Sortida:
El requisit amb els següents canvis: el requisit ha deixat de ser un requisit de tipus "nou" i passa a ser un requisit de tipus "amb patró associat" que té com a patró associat l'element rebut com a paràmetre.
Postcondició:
Deixem d'editar un requisit "nou" i passem a editar un requisit "amb patró associat", i això implica que deixem aquest model de comportament i passem al punt 2 del model de comportament "editar un requisit amb un patró associat".

<i>updateReq</i>
Context:
updateReq(req, project)
Precondició:
No existeix cap requisit amb el mateix nom a "project" ni el nom és un conjunt d'espais en blanc.
Excepcions:
Si ja existeix un requisit amb el mateix nom a "project" o el nom és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:

Postcondició:
S'actualitza al projecte el requisit rebut com a paràmetre.

Editar un requisit amb un patró associat

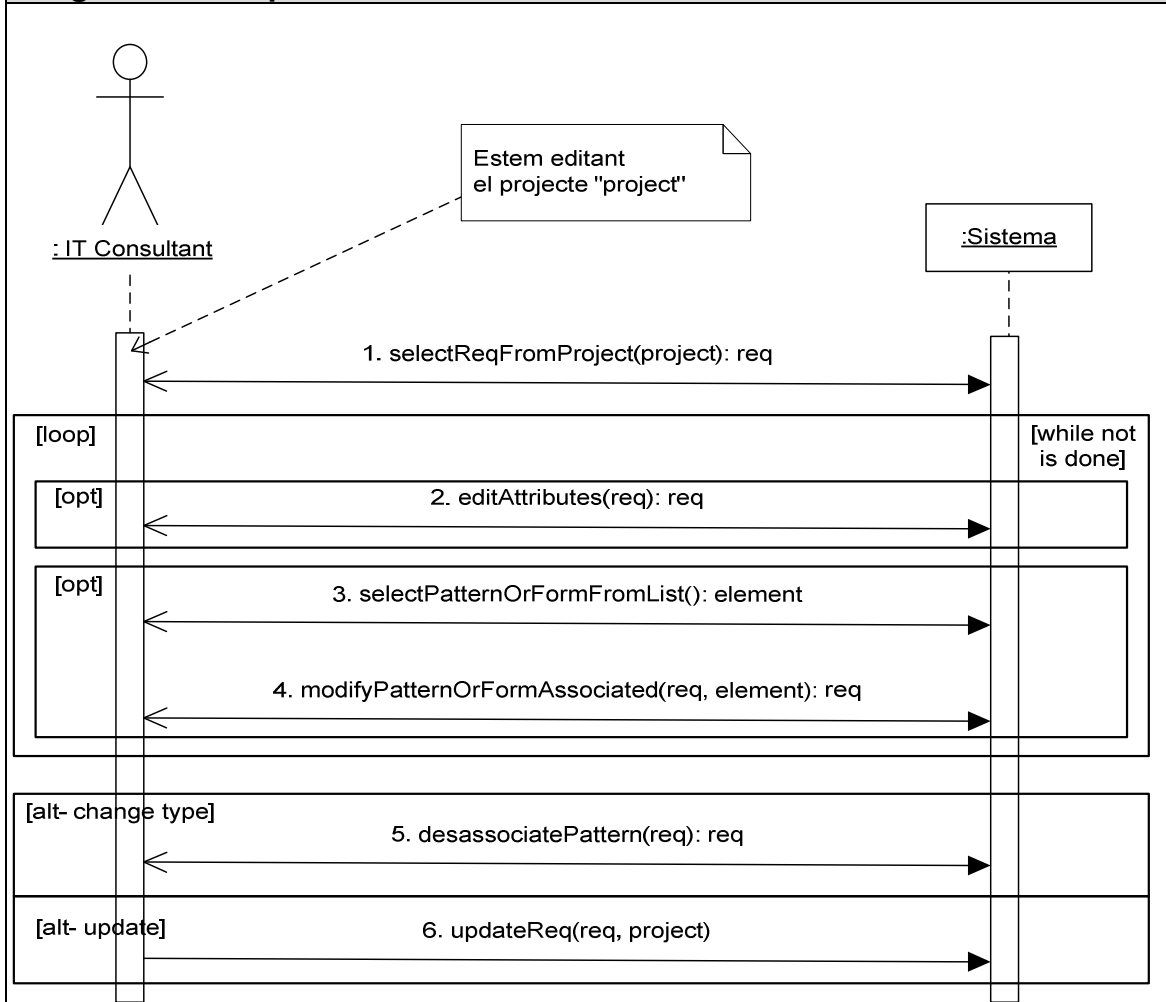
Casos d'ús coberts:

Use Case 2.4.2

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18, RI-5

Diagrama de seqüència



Contracte d'operacions

selectReqFromProject

Context:

selectReqFromProject(project): req

Precondició:

Excepcions:

Sortida:

El requisit de tipus "amb patró associat" seleccionat.

Postcondició:

<i>editAttributes</i>
Context:
<i>editAttributes</i> (req): req
Precondició:
Ni el question text ni el form text del requisit és un conjunt d'espais en blanc.
Excepcions:
Si el question text o el form text del requisit és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:
El requisit amb els seus atributs actualitzats.
Postcondició:

<i>selectPatternOrFormFromList</i>
Context:
<i>selectPatternOrFormFromList</i> (): element
Precondició:
La llista de patrons i dels seus elements està carregada.
Excepcions:

Sortida:
El patró o el requirement form seleccionat.
Postcondició:

<i>modifyPatternOrFormAssociated</i>
Context:
<i>modifyPatternOrFormAssociated</i> (req, element): req
Precondició:

Excepcions:

Sortida:
El requisit "req", però amb el patró associat modificat per "element".
Postcondició:

<i>desassociatePattern</i>	
Context:	
desassociatePattern(req): req	
Precondició:	

Excepcions:	

Sortida:	
El requisit amb els següents canvis: el requisit ha deixat de ser un requisit de tipus “amb patró associat”, i per tant deixa d’estar associat amb el patró que estava, i passar a ser un requisit de tipus “nou”.	
Postcondició:	
Deixem d’editar un requisit “amb patró associat” i passem a editar un requisit “nou”, i això implica que deixem aquest model de comportament i passem al punt 2 del model de comportament “editar un requisit nou”.	

<i>updateReq</i>	
Context:	
updateReq(req, project)	
Precondició:	
No existeix cap requisit amb el mateix nom a “project” ni el nom és un conjunt d’espais en blanc.	
Excepcions:	
Si ja existeix un requisit amb el mateix nom a “project” o el nom és un conjunt d’espais en blanc, el sistema informa de l’error.	
Sortida:	

Postcondició:	
S’actualitza al projecte el requisit rebut com a paràmetre.	

Editar un requisit instància d'un patró

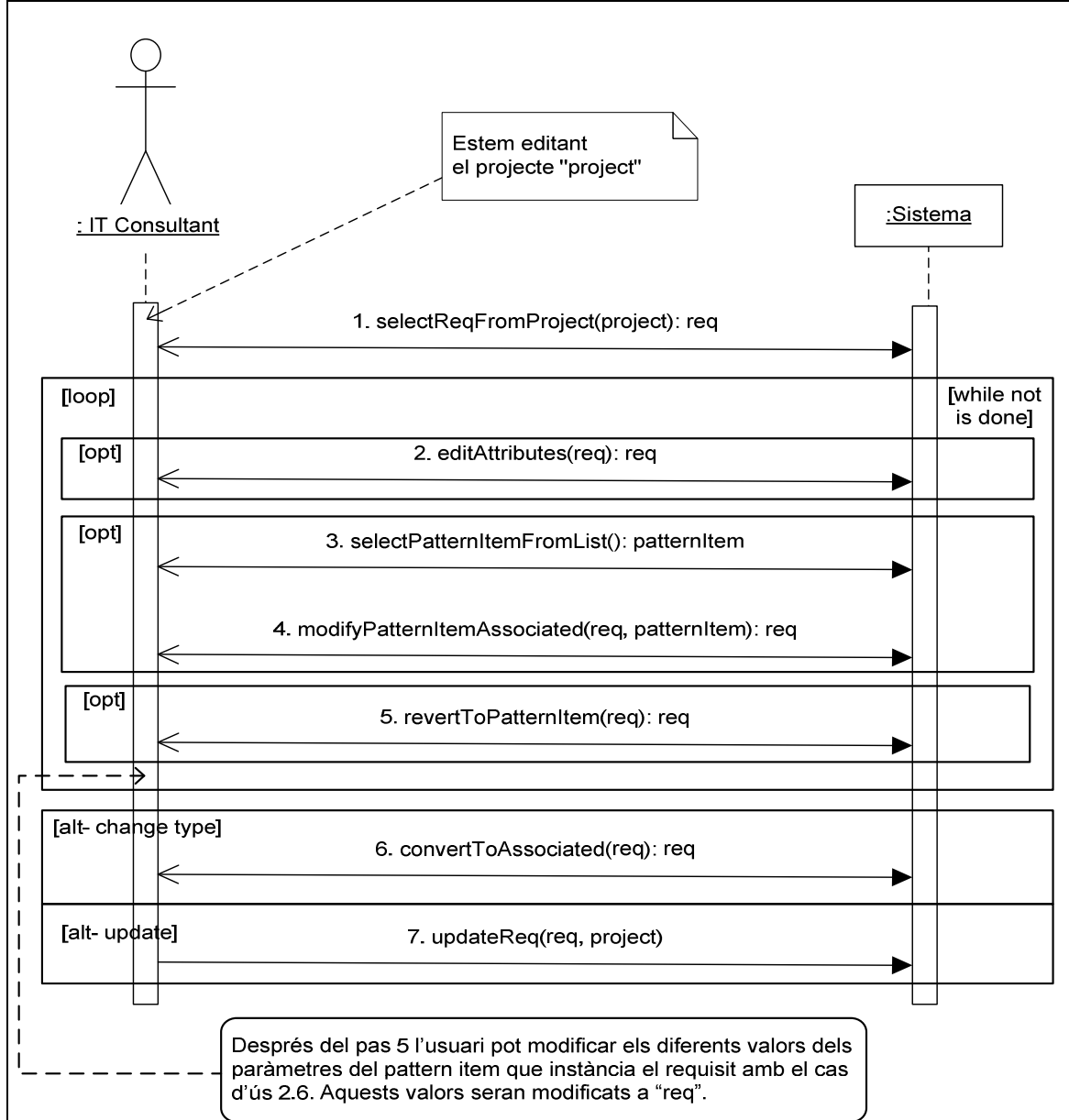
Casos d'ús coberts:

Use Case 2.4.3

Restriccions d'integritat a tenir en compte:

RI-3, RI-4, RI-16, RI-18, RI-6, RI-7, RI-8, RI-19

Diagrama de seqüència



Contracte d'operacions

selectReqFromProject

Context:

selectReqFromProject(project): req

Precondició:

Excepcions:

Sortida:
El requisit de tipus "instància d'un patró" seleccionat.
Postcondició:

<i>editAttributes</i>
Context:
editAttributes(req): req
Precondició:
Ni el question text ni el form text del requisit és un conjunt d'espais en blanc.
Excepcions:
Si el question text o el form text del requisit és un conjunt d'espais en blanc, el sistema informa de l'error.
Sortida:
El requisit amb els seus atributs actualitzats.
Postcondició:

<i>selectPatternItemFromList</i>
Context:
selectPatternItemFromList(): patternItem
Precondició:
La llista de patrons i dels seus elements està carregada.
Excepcions:

Sortida:
El pattern item seleccionat.
Postcondició:

<i>modifyPatternItemAssociated</i>
Context:
modifyPatternItemAssociated(req, patternItem): req
Precondició:

Excepcions:

Sortida:
El requisit "req", però amb el patró associat modificat per "element" i els seus question text, form text i valors substituïts pels del pattern item que el requisit ara instància.
Postcondició:

<i>revertToPatternItem</i>
Context:
revertToPatternItem(req): req
Precondició:

Excepcions:

Sortida:
El requisit amb els seu question text i form text modificats pels del pattern item que té associat.
Postcondició:

<i>convertToAssociated</i>
Context:
convertToAssociated(req): req
Precondició:

Excepcions:

Sortida:
El requisit amb els següents canvis: el requisit ha deixat de ser un requisit de tipus "instància d'un patró" i passa a ser un requisit de tipus "amb patró associat" que té com a patró associat el patró que instanciava abans del canvi.
Postcondició:
Deixem d'editar un requisit "instància d'un patró" i passem a editar un requisit "amb patró associat", i això implica que deixem aquest model de comportament i passem al punt 2 del model de comportament "editar un requisit amb un patró associat".

<i>updateReq</i>
Context:
updateReq(req, project)
Precondició:
No existeix cap requisit amb el mateix nom a "project" ni el nom és un conjunt d'espais en blanc. Ni el question text ni el form text del requisit és un conjunt d'espais en blanc. El form text té tots els noms dels paràmetres del pattern item que el requisit instància i tots els valors del requisit estan definits.
Excepcions:
Si ja existeix un requisit amb el mateix nom a "project" o el nom és un conjunt d'espais en blanc, o si el question text o el form text del requisit és un conjunt d'espais en blanc o si el form text no té el nom de tots els paràmetres del pattern item que el requisit instància, o si algun valor del requisit no està definit, el sistema informa de l'error.
Sortida:

Postcondició:
S'actualitza al projecte el requisit rebut com a paràmetre.

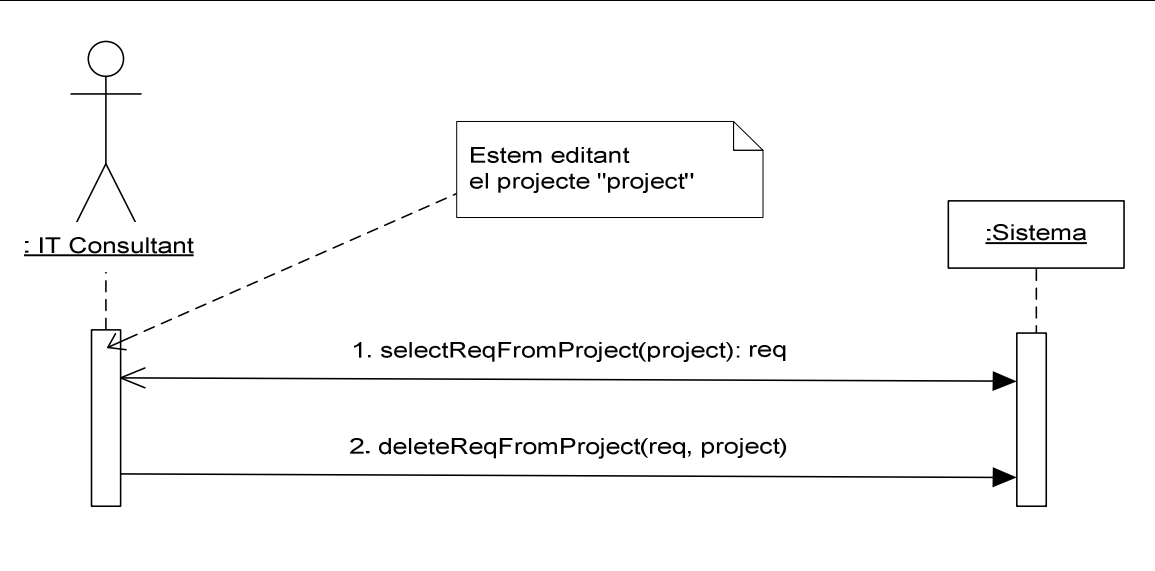
Eliminar un requisit d'un projecte

Casos d'ús coberts:

Use Case 2.5

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectReqFromProject

Context:

selectReqFromProject(project): req

Precondició:

Excepcions:

Sortida:

El requisit seleccionat.

Postcondició:

deleteReqFromProject

Context:

deleteReqFromProject(req, project)

Precondició:

"req" és un requisit de "project"

Excepcions:

Sortida:

Postcondició:

S'elimina el requisit del projecte.

Modificar el valor d'un requisit

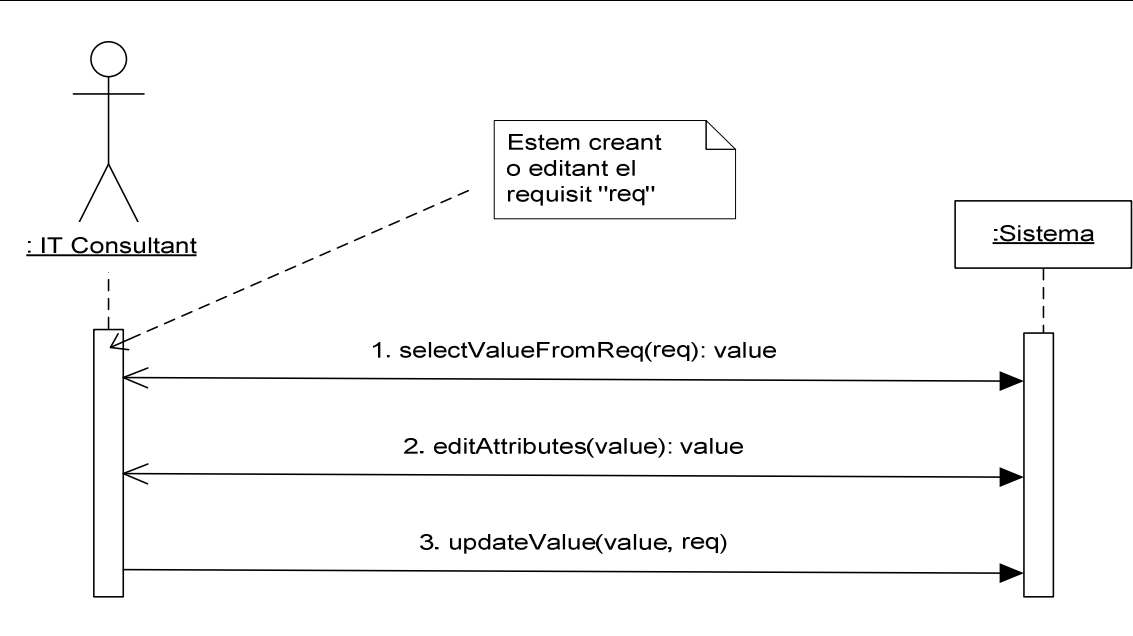
Casos d'ús coberts:

Use Case 2.6

Restriccions d'integritat a tenir en compte:

RI-9, RI-10, RI-11, RI-12, RI-13, RI-14, RI-15, RI-17

Diagrama de seqüència



Contracte d'operacions

selectValueFromReq

Context:

`selectValueFromReq(req): value`

Precondició:

Excepcions:

Sortida:

El valor seleccionat.

Postcondició:

editAttributes

Context:

`editAttributes(value): value`

Precondició:

Excepcions:

Sortida:

El valor amb els seus atributs actualitzats.

Postcondició:	

<i>updateValue</i>	
Context:	
updateReq(value, req)	
Precondició:	
Si "value" no satisfà les restriccions del paràmetre que té associat, l'usuari ha acceptat igualment modificar el valor.	
Excepcions:	

Sortida:	

Postcondició:	
S'actualitza al requisit el valor rebut com a paràmetre.	

14.4. Gestió de projectes

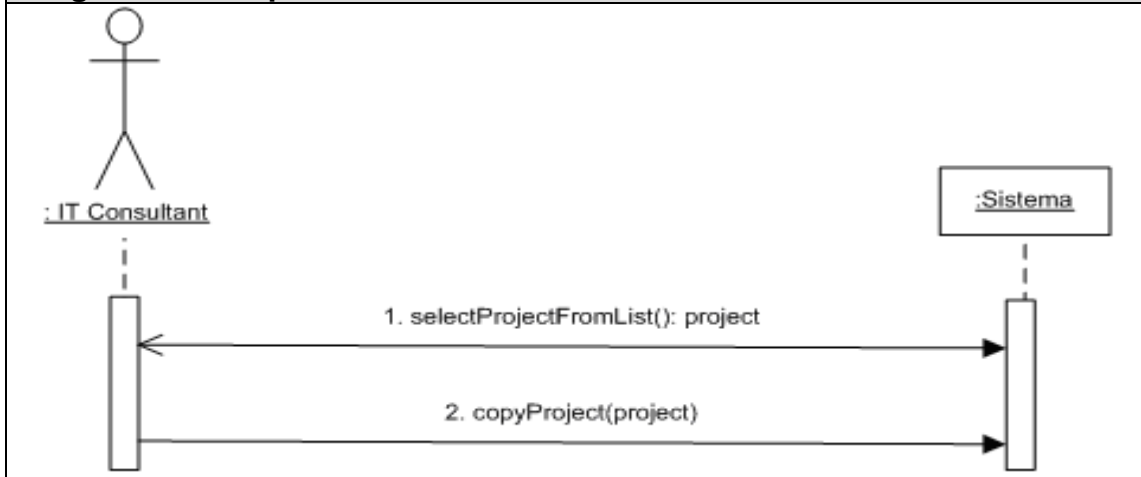
Copiar un projecte

Casos d'ús coberts:

Use Case 3.1

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectProjectFromList

Context:

`selectProjectFromList(): project`

Precondició:

La llista de projectes està carregada.

Excepcions:

Sortida:

El projecte seleccionat.

Postcondició:

copyProject

Context:

`copyProject(project)`

Precondició:

El nom de la còpia proposat pel sistema no el té cap projecte del repositori.

Excepcions:

Si el nom de la còpia ja el té un altre projecte, el sistema proposa un altre nom.

Sortida:

Postcondició:

Es copia el projecte amb el nom proposat pel sistema i es salva al repositori.

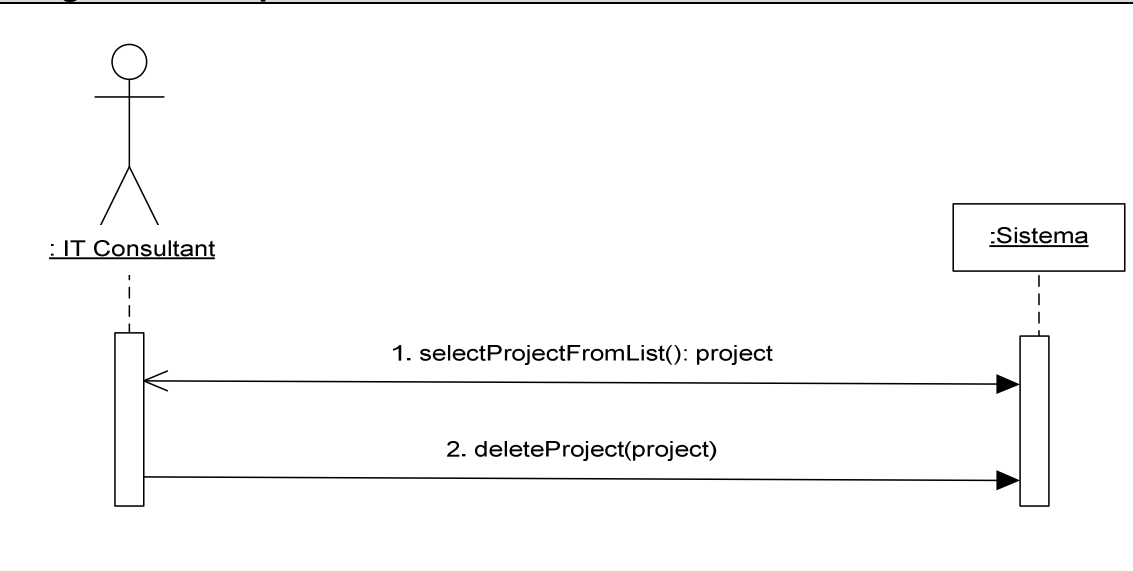
Eliminar un projecte

Casos d'ús coberts:

Use Case 3.2

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectProjectFromList

Context:

selectProjectFromList(): project

Precondició:

La llista de projectes està carregada.

Excepcions:

Sortida:

El projecte seleccionat.

Postcondició:

deleteProject

Context:

deleteProject(project)

Precondició:

Excepcions:

Sortida:

Postcondició:

El projecte s'esborra totalment del repositori.

14.5. Interoperabilitat

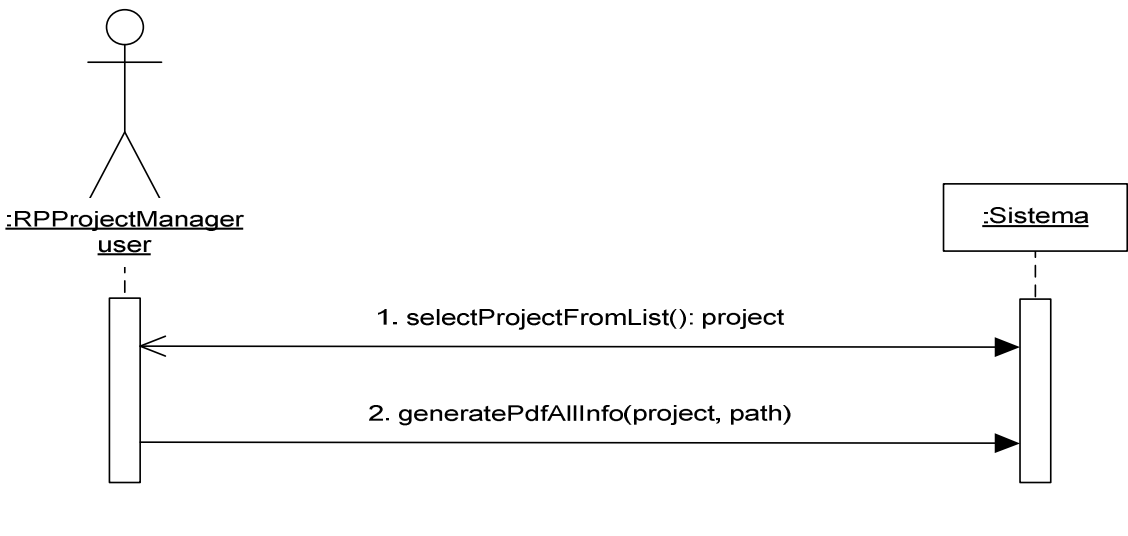
Generar pdf amb tota la informació d'un projecte

Casos d'ús coberts:

Use Case 4.1

Restriccions d'integritat a tenir en compte:

Diagrama de seqüència



Contracte d'operacions

selectProjectFromList

Context:

selectProjectFromList(): project

Precondició:

La llista de projectes està carregada.

Excepcions:

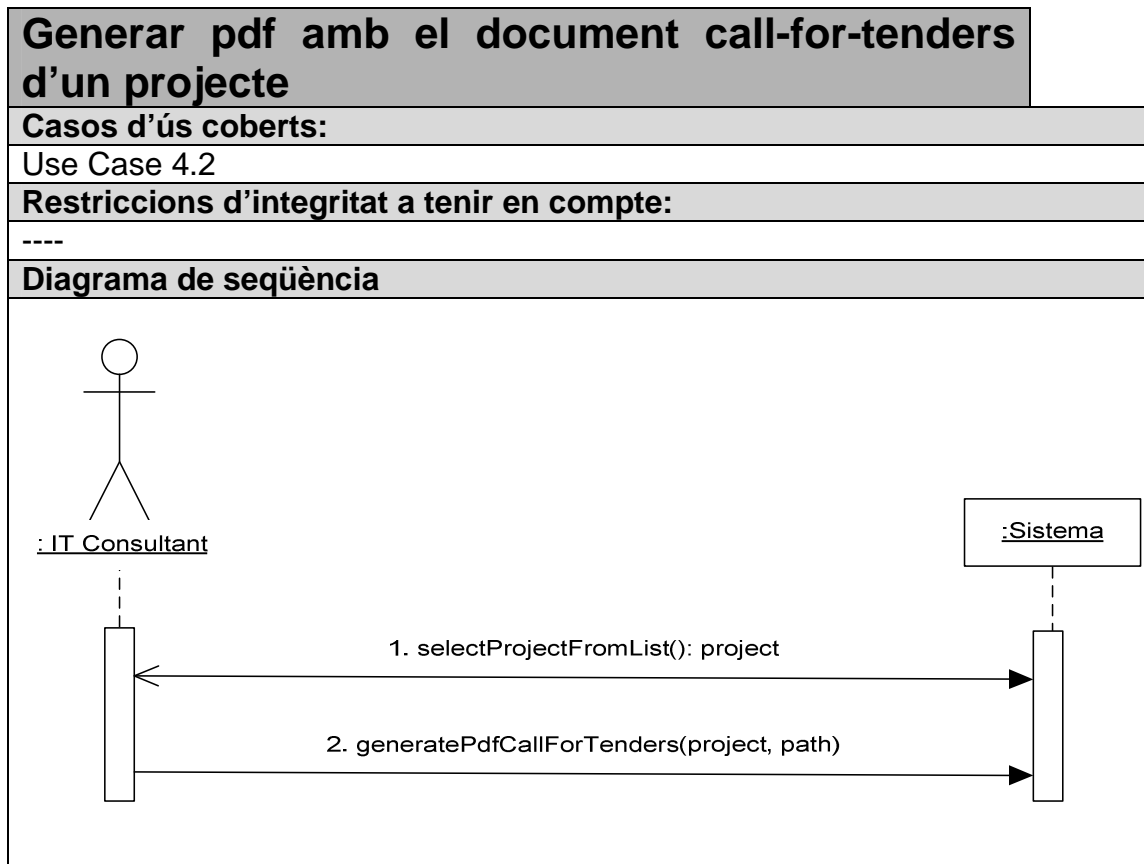
Sortida:

El projecte seleccionat.

Postcondició:

<i>generatePdfAllInfo</i>
Context:
generatePdfAllInfo(project, path)
Precondició:
El path del fitxer on guardar el pdf generat és vàlid pel sistema i no està sent usat per cap altra aplicació i, si el fitxer ja existeix, l'usuari està disposat a sobre escriure'l.
Excepcions:
Si el path del fitxer on guardar el pdf generat no és vàlid pel sistema o està sent usat per una altra aplicació, el sistema informa de l'error.
Sortida:

Postcondició:
Es crea un document pdf amb tota la informació del projecte i es guarda en el path rebut com a paràmetre.



Contracte d'operacions

<i>selectProjectFromList</i>
Context:
selectProjectFromList(): project
Precondició:
La llista de projectes està carregada.
Excepcions:

Sortida:
El projecte seleccionat.
Postcondició:

<i>generatePdfCallForTenders</i>
Context:
generatePdfCallForTenders(project, path)
Precondició:
El path del fitxer on guardar el pdf generat és vàlid pel sistema i no està sent usat per cap altra aplicació i, si el fitxer ja existeix, l'usuari està disposat a sobre escriure'l.
Excepcions:
Si el path del fitxer on guardar el pdf generat no és vàlid pel sistema o està sent usat per una altra aplicació, el sistema informa de l'error.
Sortida:

Postcondició:
Es crea un document pdf amb el contingut del document call-for-tenders del projecte i es guarda en el path rebut com a paràmetre.

15. Disseny de la Interfície

1.	Introducció	11
2.	Patrons de requisits	15
3.	Eines de Treball	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
15.1.	Introducció	168
15.2.	Disseny de les finestres	168
15.2.1.	Finestra principal	168
15.2.2.	Finestra d'edició de projectes	171
15.2.3.	Finestres d'edició de valors	180
15.2.4.	Finestres d'error	184
15.3.	Navigation Maps	184
16.	Reutilització de Components	189
17.	Base de Dades	195
18.	Document de Disseny	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

15.1. Introducció

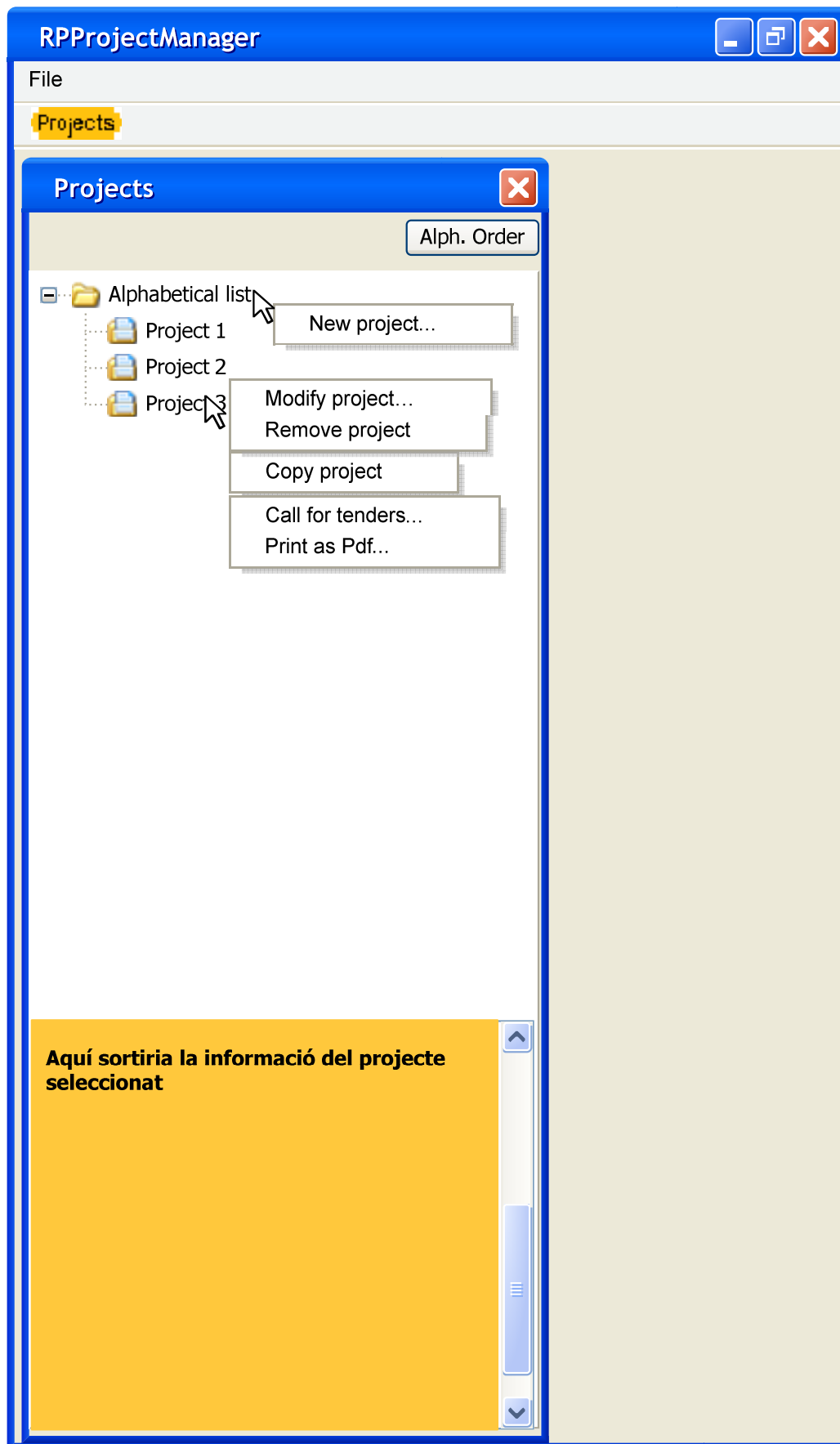
Aquest capítol té per objectiu mostrar els storyboards de l'aplicació *RProjectManager*, és a dir, el disseny de les seves finestres i diàlegs, i mostrar també com es produeix la navegació entre finestres mitjançant els navigation maps.

15.2. Disseny de les finestres

15.2.1. Finestra principal

A l'Storyboard 1 es pot veure la finestra principal de l'aplicació quan aquesta acaba de ser executada. A la barra de menús tenim l'opció comú de “File→ Exit” i també hi ha el botó “Projects”, que permet obrir i tancar la finestra de projectes.

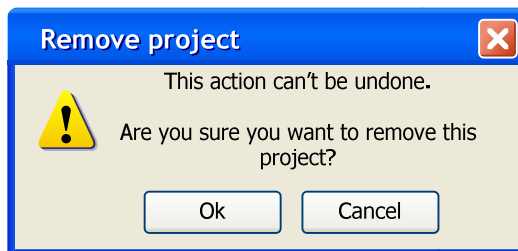
La finestra de projectes, també anomenada Projects Manager, és la finestra que conté la llista amb els noms de tots els projectes que hi ha al repositori ordenats alfabèticament. La part del fons taronja que hi ha a la finestra de projectes és un camp d'informació dinàmic que conté una petita descripció del projecte que hi ha seleccionat en l'arbre. Per últim, el botó “Alph. Order” permet reordenar la llista de projectes alfabèticament; aquest botó és necessari ja que quan es produeix una modificació en el nom del projecte aquesta llista no es torna a ordenar per tal que no es modifiqui la posició que ocupa en la llista el projecte que s'estava editant, ja que de cara a l'usuari això pot ser molt empipador.



Storyboard 1: Finestra principal

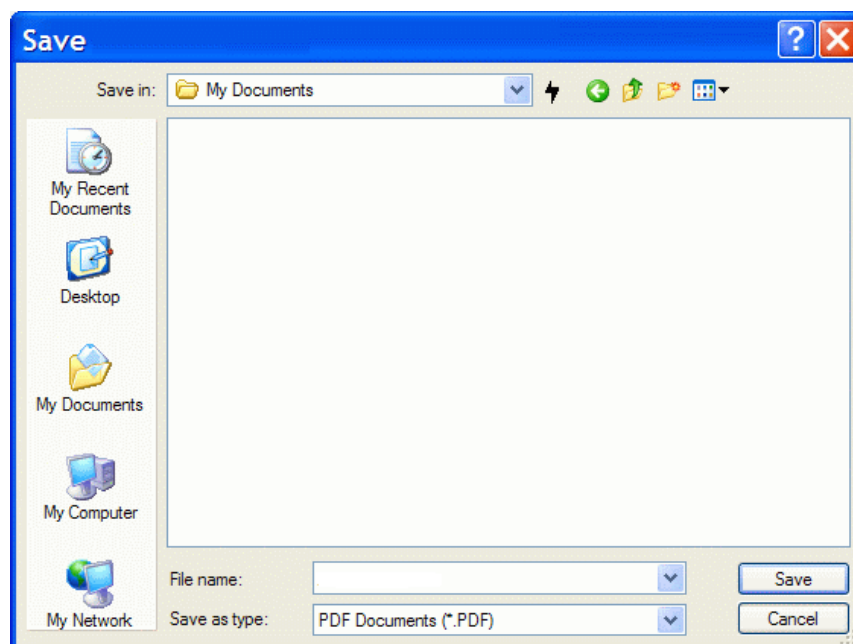
Des de la finestra de projectes, si fem clic dret a sobre d'un dels elements de l'arbre, tenim diferents accions a realitzar:

- Si fem clic dret sobre el node arrel, el d'*Alphabetical list*, tenim l'opció "New project..." que permet crear un nou projecte.
- Si fem clic dret sobre un node que representa un projecte, tenim diferents possibilitats:
 - "Modify project...": permet modificar el projecte seleccionat.
 - "Remove project...": després de demanar confirmació a l'usuari, amb l'Storyboard 2, esborra el projecte i tot el seu contingut del repositori.

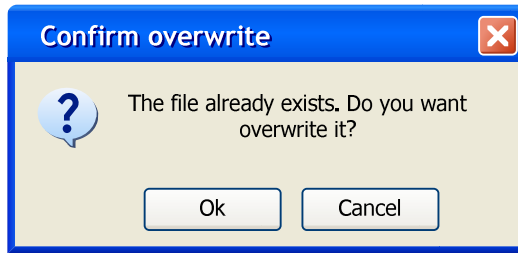


Storyboard 2: Confirmació per esborrar projecte

- "Copy project...": copia el projecte seleccionat en un nou projecte amb un nom proposat pel sistema del tipus "Copy_of_nom_projecte_seleccionat".
- "Call for tenders...": permet generar el pdf que conté la informació del document call-for-tenders del projecte seleccionat. Aquesta opció demana que seleccionem el fitxer on volem guardar el pdf (Storyboard 3) i en cas que el fitxer escollit ja existeixi demana confirmació per sobreescriure'l (Storyboard 4).



Storyboard 3: Guardar pdf



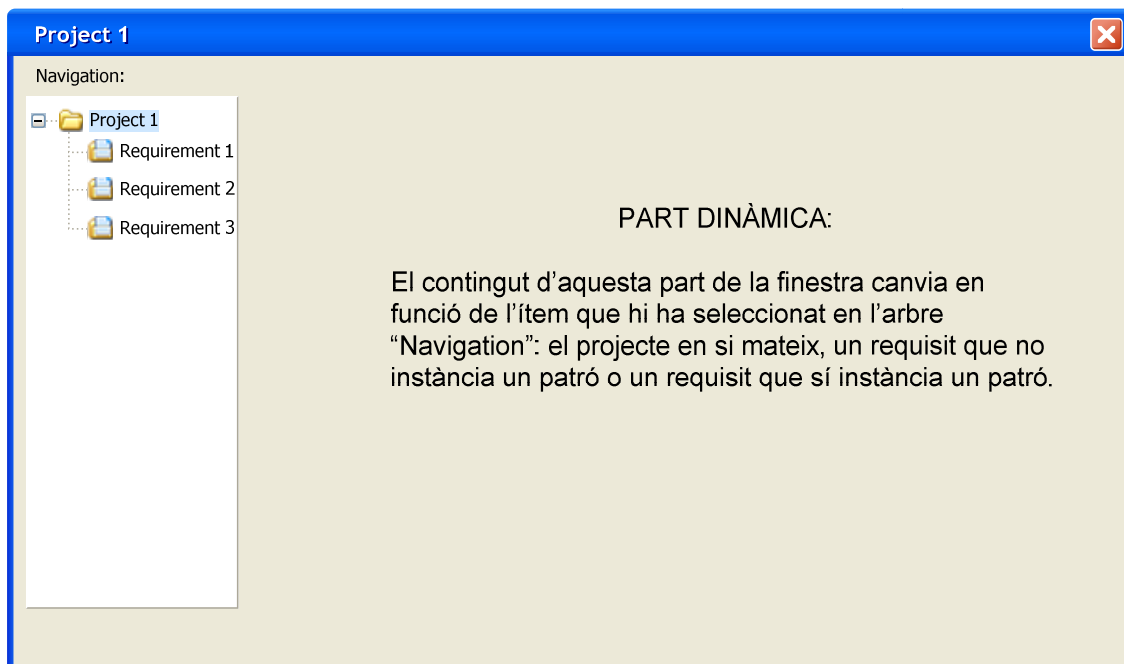
Storyboard 4: Confirmació sobreescritura fitxer

- “Print as Pdf...”: permet generar el pdf que conté tota la informació del projecte seleccionat. Aquesta opció demana que seleccionem el fitxer on volem guardar el pdf (Storyboard 3) i en cas que el fitxer escollit ja existeixi demana confirmació per sobreescriure'l (Storyboard 4).

15.2.2. Finestra d'edició de projectes

La finestra d'edició de projectes és aquella que apareix quan volem crear un nou projecte (i llavors el formulari apareix en blanc) o modificar-ne un ja existent (i llavors el formulari apareix omplert amb la informació del projecte). Tot i que el nom d'aquesta finestra és edició de projectes, des d'aquí no només es poden editar projectes, sinó que també es poden afegir requisits al projecte i esborrar o editar un dels requisits ja existents del projecte.

Aquesta finestra té una part dinàmica i una part estàtica. La part estàtica fa referència a la part que no canvia mai, estiguem editant un projecte o un requisit de qualsevol tipus (aquesta part es pot veure a l'Storyboard 5). La part dinàmica canvia en funció de l'ítem que hi hagi seleccionat en l'arbre “Navigation” d'aquesta finestra, que pot ser un projecte, un requisit que no instanciï un patró o un requisit que sí instanciï un patró.

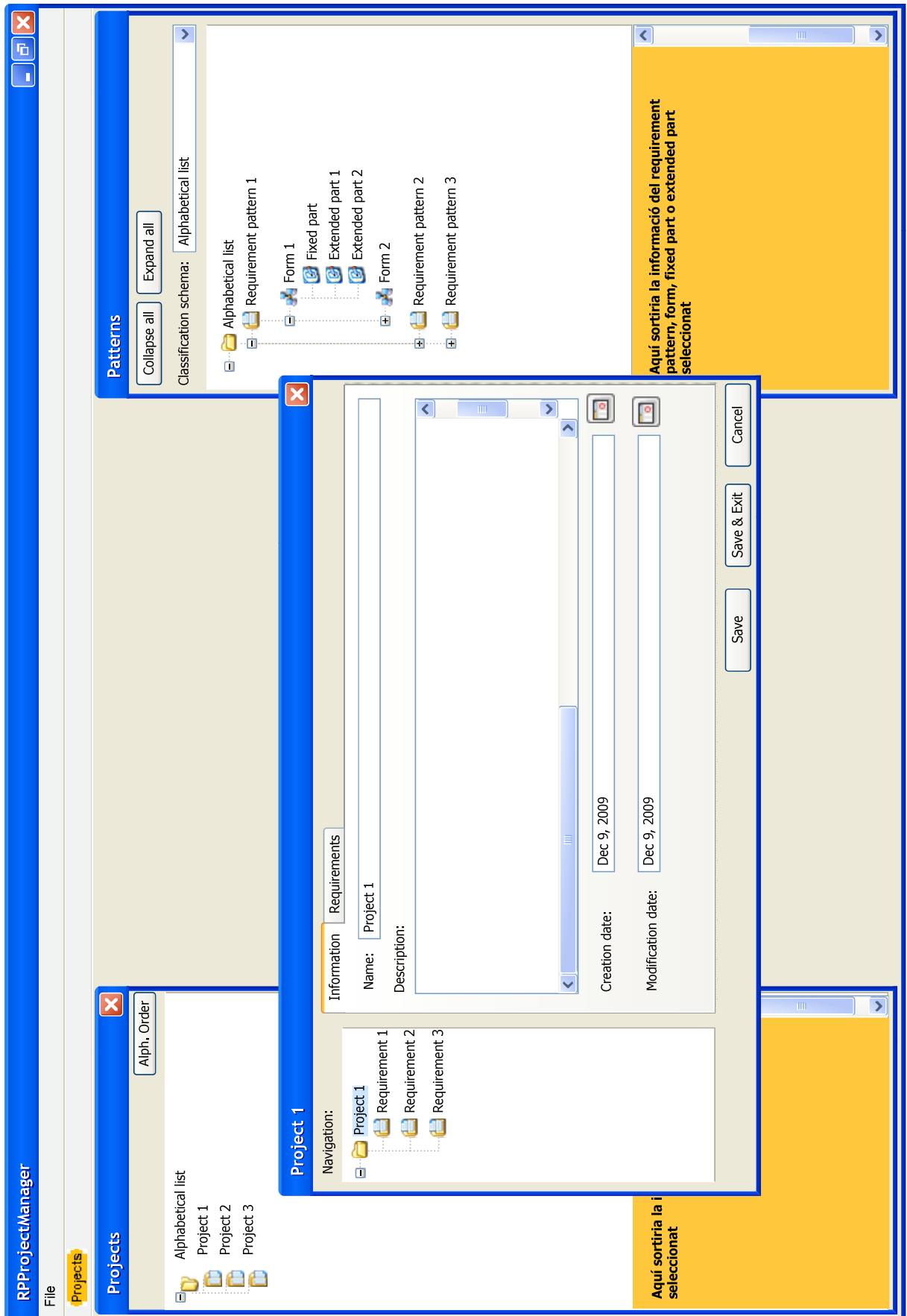


Storyboard 5: Finestra d'edició de projectes, part estàtica

La part de “Navigation” de la finestra, com el seu propi nom indica, ens permet navegar pel contingut del projecte d’una forma àgil i simple: seleccionant un ítem d’aquest arbre (ja sigui el projecte en si mateix o un dels seus requisits) passem a editar-lo directament, la qual cosa significa que la part dinàmica de la finestra canvia i s’adapta al nou ítem seleccionat.

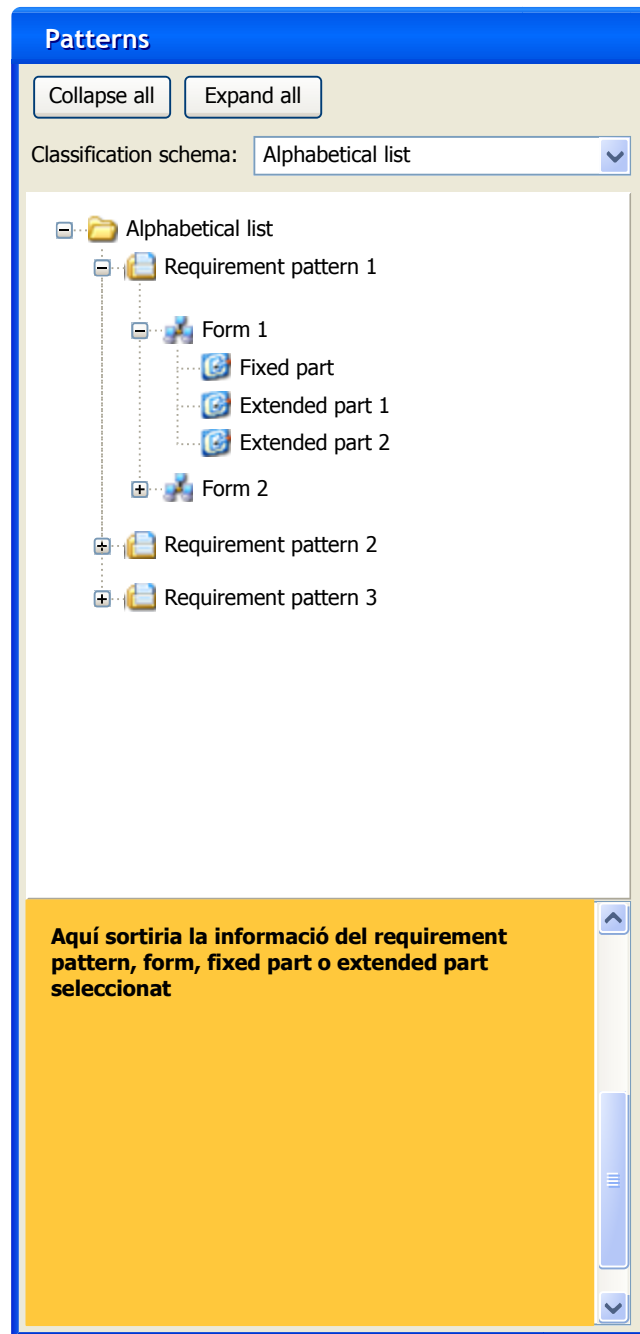
El botons “Save” i “Save&Exit” ens permeten guardar el projecte i tots els seus requisits al repositori. La diferència entre un i l’altre és que amb “Save” la finestra d’edició continua oberta i amb “Save&Exit” la finestra d’edició es tanca. Amb el botó “Cancel” tanquem la finestra d’edició sense guardar els canvis realitzats.

Cal afegir que sempre que obrim una finestra d’edició de projectes s’obre també una finestra de patrons, també anomenada Patterns Manager, al costat dret, com es pot veure a l’Storyboard 6.



Storyboard 6: Vista general del programa

Com ja hem dit, la finestra de patrons, o Patterns Manager, s'obre quan obrim una finestra d'edició de projectes i es tanca quan tanquem aquesta última. A l'Storyboard 7 podem veure amb més detall la finestra de patrons.



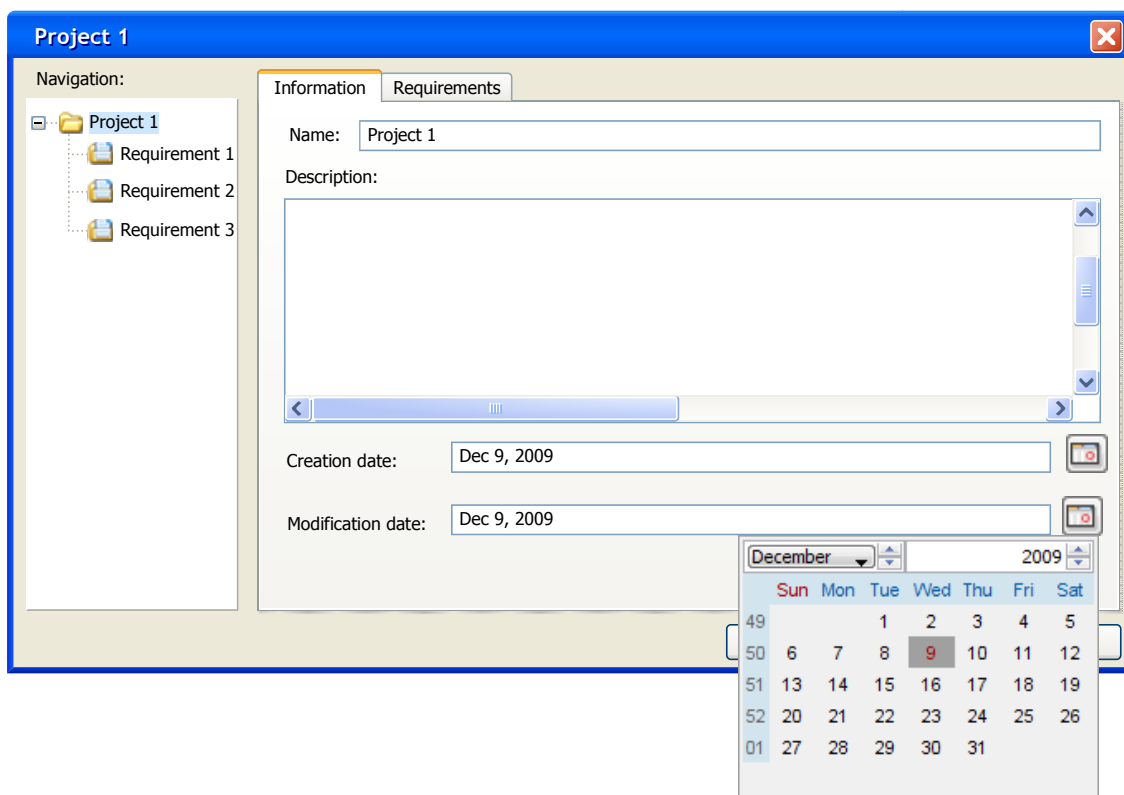
Storyboard 7: Patterns Manager (finestra de patrons)

Aquesta finestra conté la llista amb els noms de tots els patrons que hi ha al repositori junt amb els seus elements (forms, fixed i extended parts) ordenats segons el classification schema seleccionat al desplegable. La part del fons taronja que hi ha a la finestra és un camp d'informació dinàmic que conté una petita descripció de l'element que està seleccionat en l'arbre. Els botons "Collapse All" i "Expand All" permeten col·lapsar i expandir tot el contingut de l'arbre d'aquesta finestra. El desplegable "Classification Schema" permet seleccionar quin classification schema volem per visualitzar l'arbre de patrons.

A continuació passem a explicar quin és el contingut de la part dinàmica de la finestra d'edició de projectes en funció de l'ítem seleccionat en l'arbre "Navigation".

15.2.2.1 Finestra d'edició de projectes: editant un projecte

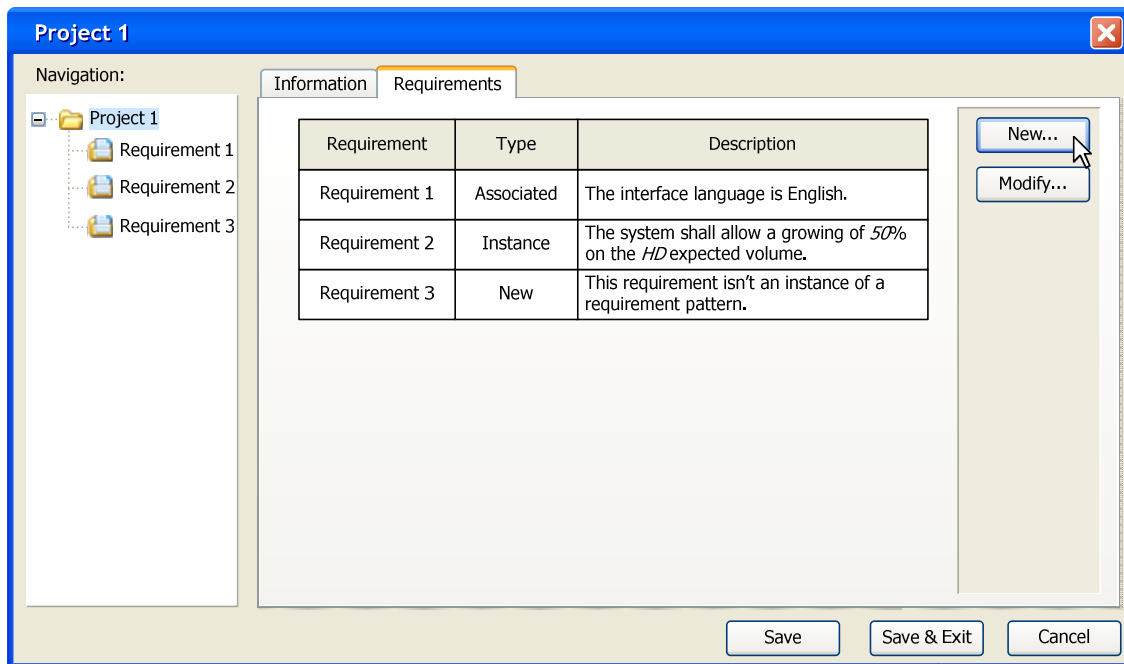
A l'Storyboard 8 podem veure la finestra d'edició de projectes quan l'ítem seleccionat a l'arbre "Navigation" és el projecte en si mateix (l'arrel de l'arbre). Des de la pestanya "Information", la que es veu en l'Storyboard 8, podem editar el nom del projecte i la seva descripció. També podem editar la seva data de creació i modificació, bé entrant directament la data al camp editable corresponent o bé seleccionant una data al calendari desplegable que apareix quan es prem el botó que hi ha al costat dels camps.



Storyboard 8: Finestra d'edició de projectes, editant projecte, pestanya "Information"

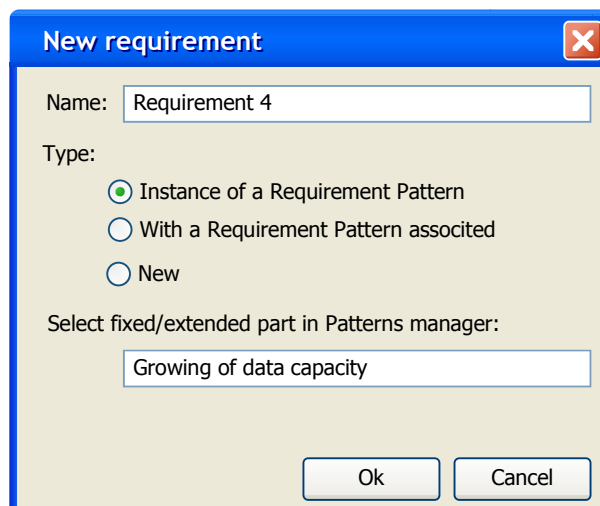
Des de la pestanya "Requirements", la que es veu en l'Storyboard 9, podem veure quins són els requisits que formen part del projecte i podem amb el botó:

- "New...": crear nous requisits.
- "Modify...": passar a editar el requisit que hi hagi seleccionat a la taula, la qual cosa significa que la part dinàmica de la finestra canvia per tal de poder editar el requisit i l'ítem seleccionat a l'arbre "Navigation" deixa de ser el projecte i passa a ser el requisit que s'està editant.
- "Remove": esborra el requisit que hi hagi seleccionat a la taula.



Storyboard 9: Finestra d'edició de projectes, editant projecte, pestanya "Requirements"

Quan passem a crear un nou requisit amb el botó "New..." ens apareix un petit diàleg on introduïm el nom del nou requisit i de quin tipus volem que sigui aquest. Si volem que el nou requisit sigui de tipus "instància d'un patró", se'ns demanarà que seleccionem a la finestra de patrons una fixed o extended part per instanciar (Storyboard 10). Si volem, per contra, que el nou requisit tingui "un patró associat", se'ns demanarà que seleccionem a la finestra de patrons el patró o form que volem associar al requisit (Storyboard 11). Si, per últim, volem que el requisit sigui un requisit "nou", no se'ns demanarà cap informació més (Storyboard 12).



Storyboard 10: Finestra nou requisit, tipus instància de patró

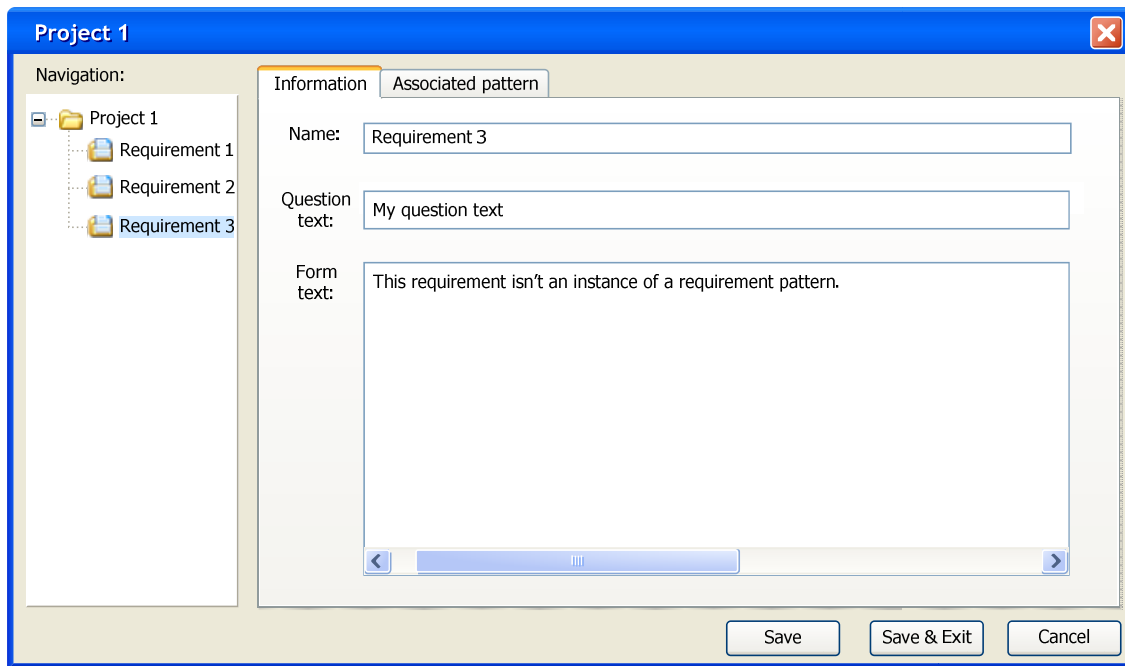
Storyboard 11: Finestra nou requisit, tipus "amb patró associat"

Storyboard 12: Finestra nou requisit, tipus "nou"

Un cop li donem a “Ok” des de qualsevol dels tres diàlegs anteriors, aquest desapareix i la part dinàmica de la finestra d’edició de projectes canvia per tal de poder editar el nou requisit i l’ítem seleccionat a l’arbre “Navigation” d’aquesta deixa de ser el projecte i passa a ser el nou requisit que s’està editant.

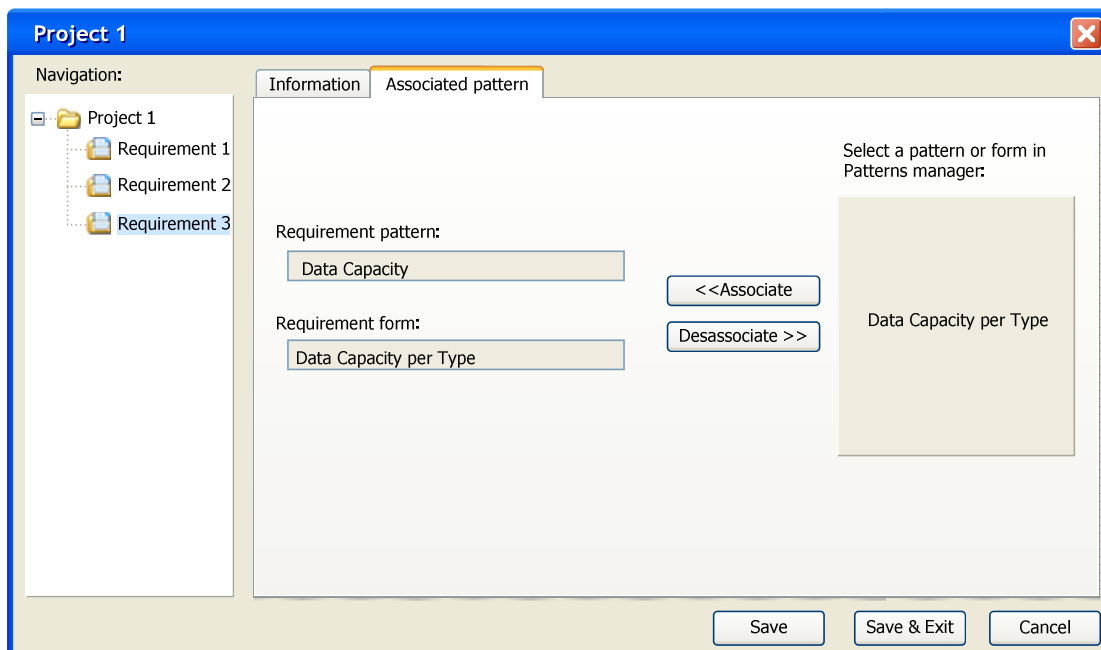
15.2.2.2 Finestra d’edició de projectes: editant un requisit que no instància un patró

Els requisits que no instancien un patró són aquells que són de tipus “nou” o que tenen “un patró associat”. A l’Storyboard 13 podem veure la finestra d’edició de projectes quan l’ítem seleccionat a l’arbre “Navigation” és un requisit que no instància un patró. Des de la pestanya “Information”, la que es veu en l’Storyboard 13, podem editar el nom, el question text i el form text del requisit.



Storyboard 13: Finestra d'edició de projectes, editant requisit que no instància un patró, pestanya "Information"

Des de la pestanya “Associated pattern”, la que es veu en l’Storyboard 14, podem veure, en cas que el requisit tingui “un patró associat”, quin és aquest. En cas que el requisit sigui de tipus “nou” els camps del formulari apareixeran en blanc.



Storyboard 14: Finestra d'edició de projectes, editant requisit que no instància un patró, pestanya "Associated pattern"

Amb el botó “Desassociate” podem fer que un requisit “amb patró associat” deixi de tenir un patró associat i passi a ser un requisit “nou”. Amb el botó “Associate”, seleccionant prèviament un patró o form a la finestra de patrons, podem:

- Si el requisit és de tipus “nou”, podem associar-li el patró seleccionat i fer que el requisit deixi de ser de tipus “nou” i passi a ser un requisit “amb patró associat”.
- Si el requisit és de tipus “amb patró associat”, podem modificar quin és el patró que té associat.

15.2.2.3 Finestra d'edició de projectes: editant un requisit que instància un patró

Quan creem un requisit que instància un patró, el question text, el form text i els valors d'aquest passen a ser els de la fixed/extended part que el requisit instància. A més, si un paràmetre de la fixed/extended part que el requisit instància té valor per defecte, s'assignarà inicialment el valor per defecte al valor del requisit.

A l'Storyboard 15 podem veure la finestra d'edició de projectes quan l'ítem seleccionat a l'arbre “Navigation” és un requisit que instància un patró. Des de la pestanya “Information”, la que es veu en l'Storyboard 15, podem editar el nom, el question text i el form text del requisit. Amb el botó “Revert to template” podem fer que el question text i el form text del requisit passin a ser exactament els mateixos que els de la fixed/extended part que instància el requisit. Amb el botó “Modify value...” podem modificar el valor associat al paràmetre que hi hagi seleccionat a la taula (per més informació sobre la modificació de valors veure el següent apartat 15.2.3 Finestres d'edició de valors). Per últim, amb el botó “Convert to associated requirement” podem fer que el requisit deixi de ser “una instància d'un patró” i passi a ser un requisit “amb patró associat”, tenint com a patró associat el patró que abans instanciava, la qual cosa significa que la part dinàmica de la finestra canvia per tal d'adaptar-se al nou tipus de requisit que s'està editant.

Project 1

Navigation:

- Project 1
 - Requirement 1
 - Requirement 2**
 - Requirement 3

Information Associated Pattern

Name: Requirement 2

Question text: This requirement isn't an instance of a requirement pattern.

Form text: The system shall allow a growing of %volPercentage% on the %resource% expected volume.

Parameter Values:

Parameter name	Parameter type	Value
volPercentage	float	50
resource	string	HD

Revert to template

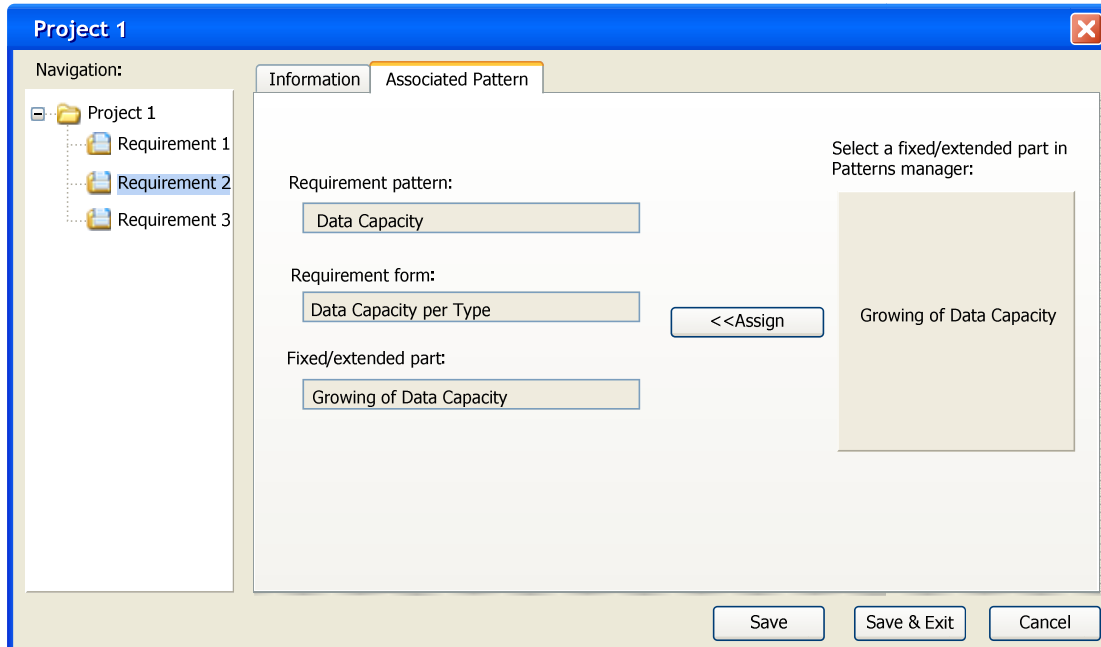
Modify value...

Convert to associated requirement

Save Save & Exit Cancel

Storyboard 15: Finestra d'edició de projectes, editant requisit que instància un patró, pestanya "Information"

Des de la pestanya “Associated pattern”, la que es veu en l’Storyboard 16, podem veure quina és la fixed/extended part que el requisit instància i modificar quina és seleccionant una nova fixed/extended part a la finestra de patrons i prement el botó “Assign”. El fet de modificar quina és la fixed/extended part que instància un requisit fa que tant el question text com el form text com els valors del requisit es modifiquin pels de la fixed/extended part que ara s’instància.



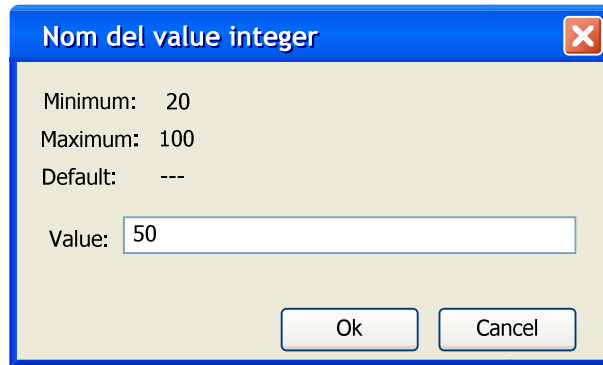
Storyboard 16: Finestra d'edició de projectes, editant requisit que instància un patró, pestanya "Associated pattern"

15.2.3. Finestres d'edició de valors

A l'anterior Storyboard 15, que es correspon amb la finestra d'edició de projectes quan l'ítem seleccionat a l'arbre “Navigation” és un requisit que instància un patró, a la pestanya “Information” tenim el botó “Modify value...” que ens permet modificar el valor de la fila que hi ha seleccionada a la taula. En funció del tipus d'aquest valor ens apareixerà una finestra per editar o una altra. A continuació mostrem quins poden ser els tipus dels valors i les finestres que permeten modificar-los.

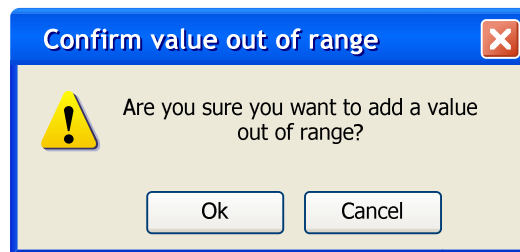
VALOR TIPUS INTEGER

A l'Storyboard 17 podem veure la finestra que ens permet modificar un valor de tipus integer. A la part de dalt ens apareix la informació del paràmetre que el valor té associat (quin és el seu mínim, màxim i valor per defecte, si és que en té) i després el camp per entrar el valor que l'usuari desitgi.



Storyboard 17: Finestra d'edició de valors tipus integer

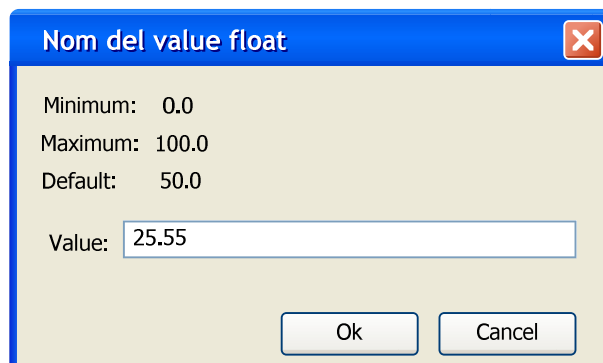
En cas que el valor entrat per l'usuari sigui un valor fora de rang (no estigui comprès entre el mínim i el màxim) apareixerà un diàleg com l'Storyboard 18 demanant confirmació a l'usuari de si realment vol afegir un valor que no compleixi les restriccions.



Storyboard 18: Diàleg confirmació valor fora de rang

VALOR TIPUS FLOAT

La finestra d'edició de valors float, que podem veure a l'Storyboard 19, és molt similar a la de tipus integer. A la part de dalt, ens apareix la informació associada al seu paràmetre (quin és el seu mínim, màxim i valor per defecte, si és que en té) i després el camp per entrar el valor que l'usuari desitgi.

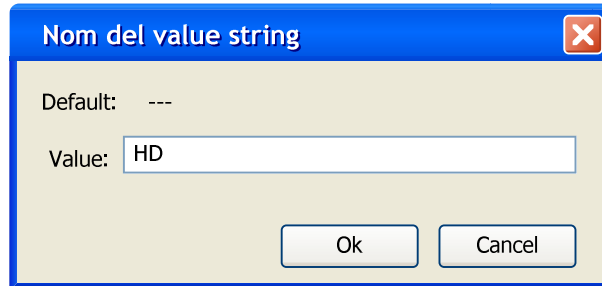


Storyboard 19: Finestra d'edició de valors tipus float

En cas que el valor entrat per l'usuari sigui un valor fora de rang (no estigui comprès entre el mínim i el màxim) apareixerà un diàleg com l'anterior Storyboard 18 demanant confirmació a l'usuari de si realment vol afegir un valor que no compleixi les restriccions.

VALOR TIPUS STRING

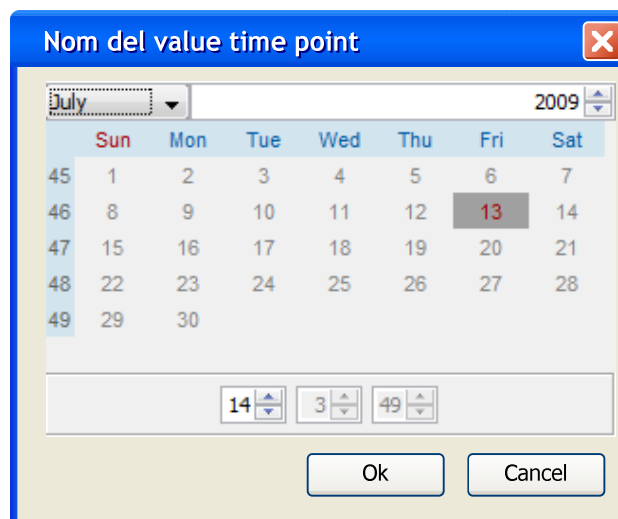
A l'Storyboard 20 podem veure la finestra que ens permet modificar un valor de tipus string. A la part de dalt, ens apareix el valor per defecte associat al paràmetre del valor, si és que en té, i a sota el camp per entrar el valor que l'usuari desitgi.



Storyboard 20: Finestra d'edició de valors tipus string

VALOR TIPUS TIME POINT

A l'Storyboard 21 podem veure la finestra que ens permet modificar un valor de tipus time point. En la finestra apareixeran actius només els camps que el paràmetre associat al valor consideri. Per exemple, si el paràmetre d'un valor té associada una mètrica de tipus time point que només té actius els camps “any”, “mes” i “dia”, el paràmetre només considerarà aquests tres camps i a la finestra d'edició del valor només apareixeran actius els camps “any”, “mes” i “dia”, i els camps “hora”, “minut” i “segon” romandran inactius.

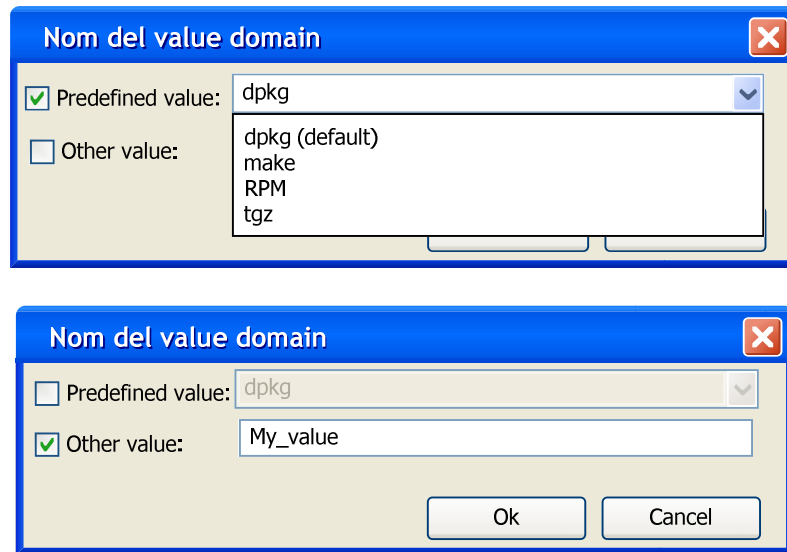


Storyboard 21: Finestra d'edició de valors tipus time point

VALOR TIPUS DOMAIN

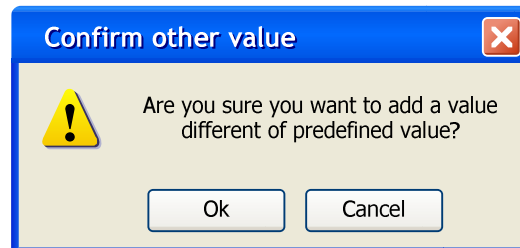
A l'Storyboard 22 podem veure la finestra que ens permet modificar un valor de tipus domain. A la part de “Predefined value” ens apareixen en un desplegable els possibles valors del paràmetre associat, indicant quin és el valor per defecte, si és que en té. A la

part de “Other value” tenim un camp editable que permet afegir qualsevol valor que l’usuari desitgi.



Storyboard 22: Finestra d'edició de valors tipus domain

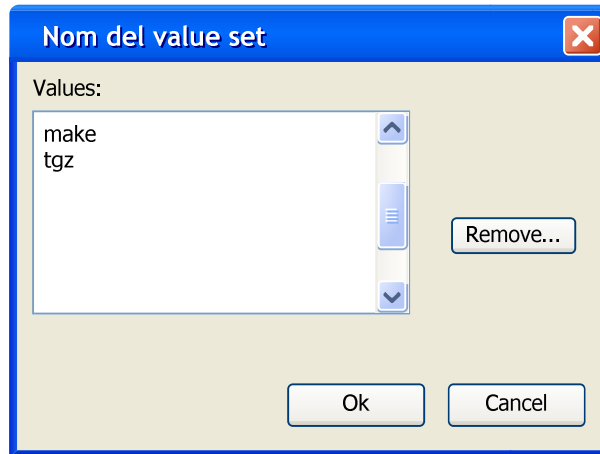
En cas que el valor entrat per l’usuari sigui un “Other value” apareixerà un diàleg com l’Storyboard 23 demanant confirmació a l’usuari de si realment vol afegir un valor que no és un dels possibles valors i que per tant no satisfà les restriccions.



Storyboard 23: Confirmació valor diferent dels valors possibles

VALOR TIPUS SET

A l’Storyboard 24 podem veure la finestra que ens permet modificar un valor de tipus set. A la llista “Values” ens apareixen quins són els valors que formen part actualment del conjunt. Amb el botó “Add...” podem afegir nous valors al conjunt: quan es prem aquest botó apareix una nova finestra, idèntica a alguna de les finestres de modificar anteriors, en funció de quin sigui el tipus simple del conjunt (integer, float, string, time point o domain), que permet afegir un nou valor al conjunt. Amb el botó “Remove” esborrem del conjunt el valor seleccionat de la llista “Values”.



Storyboard 24: Finestra d'edició de valors tipus set

15.2.4. Finestres d'error


Les finestres d'error són aquelles que apareixen quan es prem “Ok”, “Save” o “Save&Exit” en algun dels formularis anteriors, però algun valor introduït per l'usuari és incorrecte o s'ha produït algun error a l'intentar realitzar l'acció associada al botó en qüestió. A l'Storyboard 25 podem veure la plantilla del que seria una finestra d'error.



Storyboard 25: Finestra d'error

15.3. Navigation Maps

Els navigation maps tenen per objectiu mostrar l'estructura sencera de la part d'interfície gràfica d'un projecte; per aconseguir-ho, es mostra quines són les finestres que conformen el projecte, sense detalls, i les connexions que existeixen per arribar d'una finestra a una altra. Els navigation maps són útils per veure clarament i d'una forma organitzada com s'ha de produir la navegació entre finestres.

A les següents pàgines es pot veure el navigation map referent a l'aplicació *RPPProjectManager*; per facilitar la seva lectura, aquest s'ha dividit en quatre parts. Als navigation maps 2, 3 i 4 es pot trobar el símbol  (similar a l'herència en UML) que indica que totes les finestres que “hereten” de la finestra “pare” tenen també, a més de les seves pròpies transicions, totes les del seu “pare”; s'ha pres aquesta decisió per tal de reduir el nombre de navegacions en els navigation maps i fer així que aquests siguin més fàcils d'entendre i seguir.

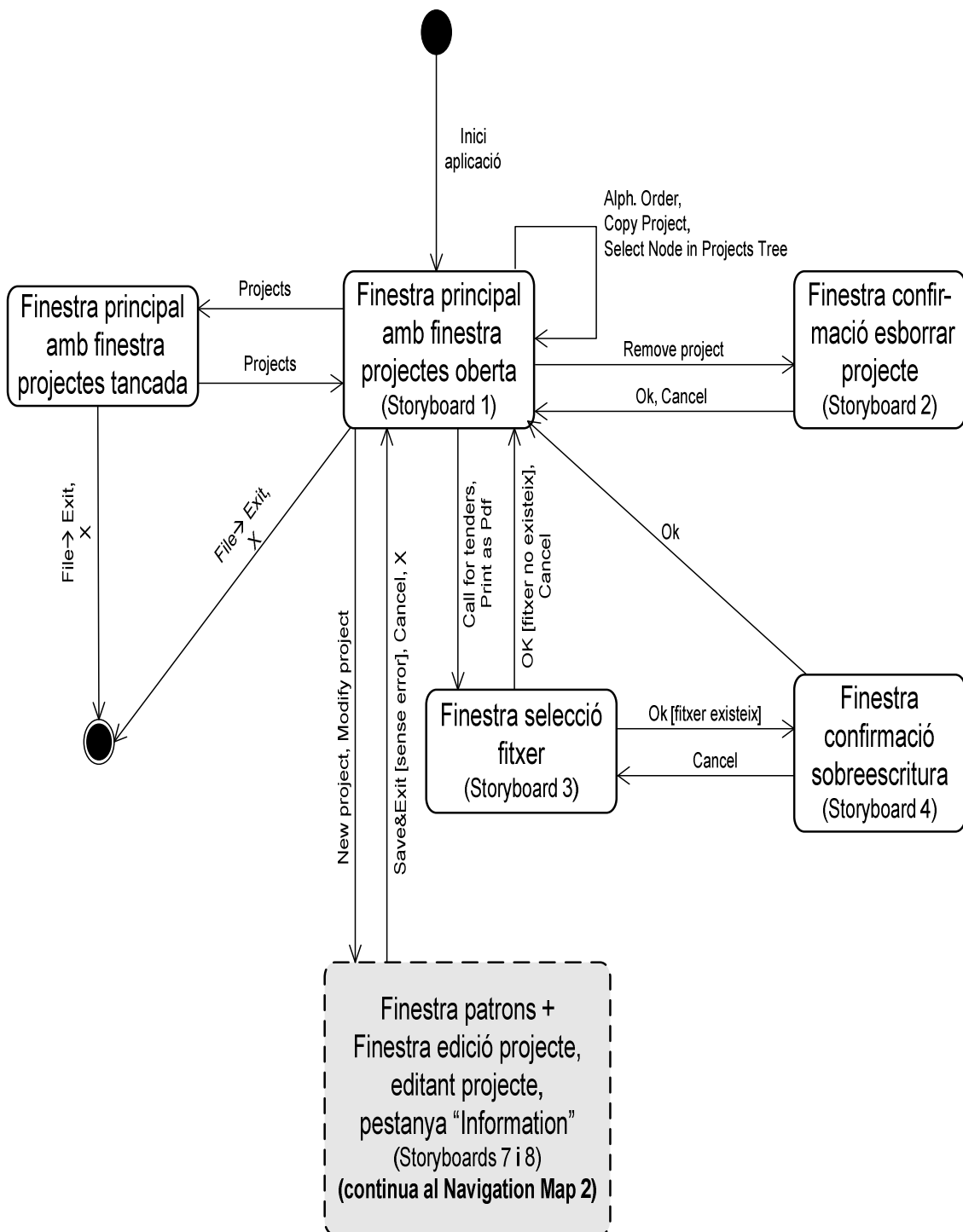


Figura 21: Navigation Map 1

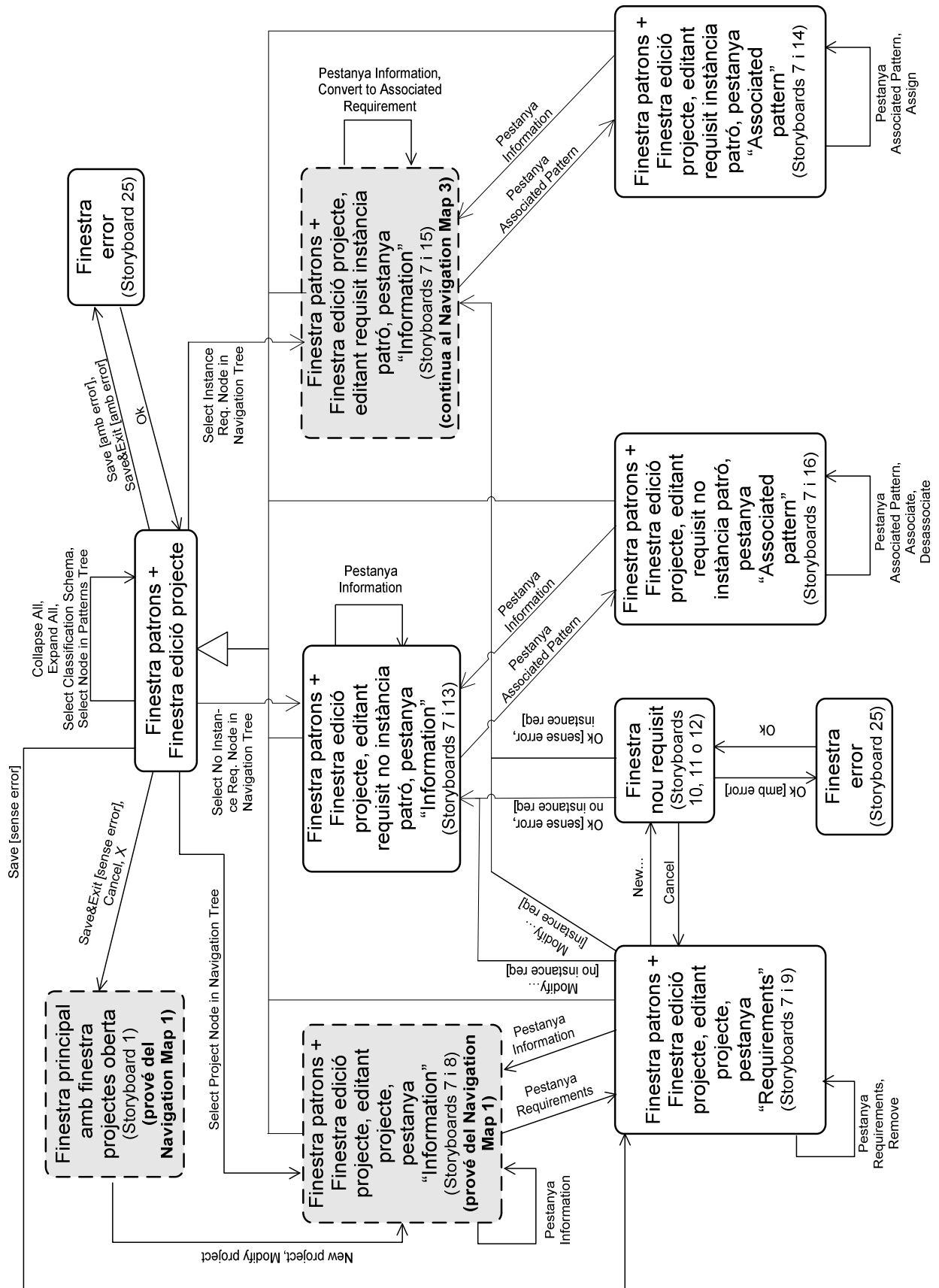


Figura 22: Navigation Map 2

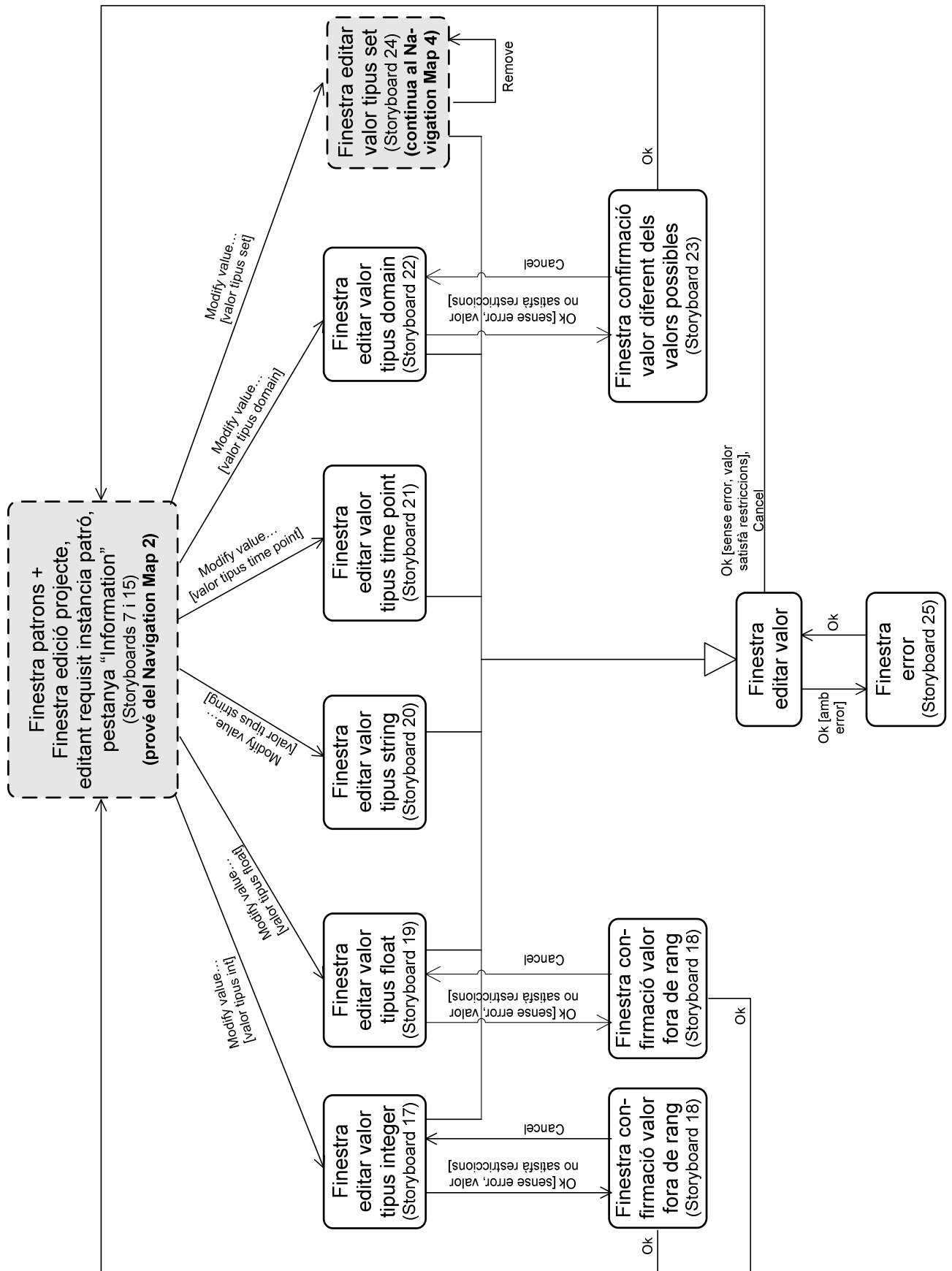


Figura 23: Navigation Map 3

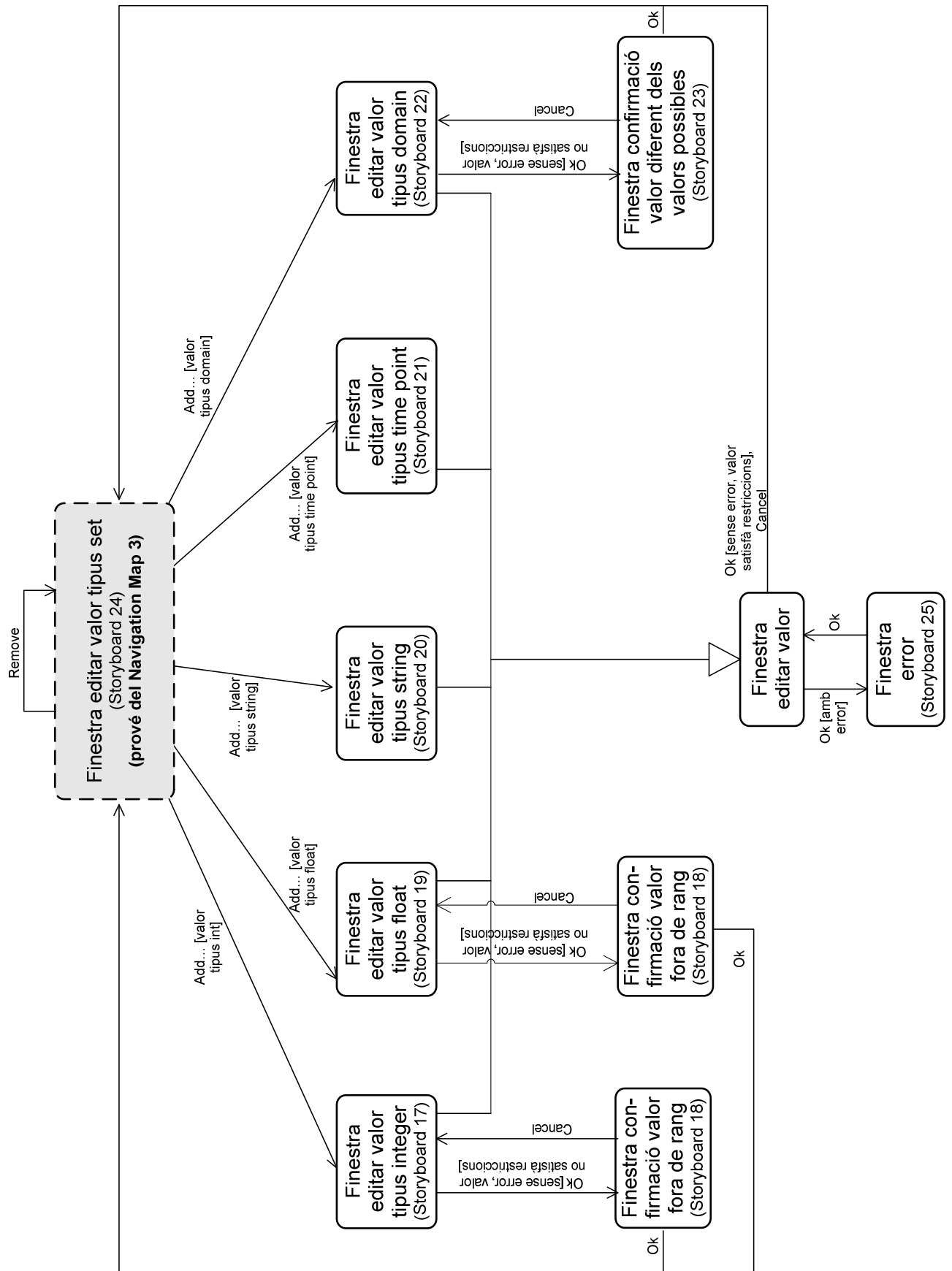


Figura 24: Navigation Map 4

16. Reutilització de Components

1. Introducció.....	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79	
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
16.1. Introducció	190
16.2. Reutilització dels components per la finestra principal	190
16.2.1. GenericMainFrame.....	190
16.2.2. InternalFrameGeneric	191
16.2.3. GlobalValues	191
16.2.4. SliderAccessDB.....	192
16.3. Reutilització dels components per a la visualització dels arbres.....	192
16.3.1. GenericTreeCellRenderer	192
16.3.2. SortedMutableTreeNode	193
16.3.3. GenericComparator	193
16.4. Reutilització dels components per la finestra d'edició de projectes	193
16.4.1. OkCancelDialog	194
16.4.2. GenericObjectTabs	194
16.4.3. DocumentWithLimitCharacters	194
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager	265

16.1. Introducció

En aquest capítol s'explicarà quins components s'han reutilitzat pel desenvolupament de la 1^a versió de l'aplicació *RPPProjectManager*, la majoria d'ells provinents de la 2^a versió de l'aplicació *RPToolManager*. La reutilització ha estat motivada pel fet d'intentar mantenir les aplicacions *RPToolManager* i *RPPProjectManager* amb una interfície gràfica similar, per tal de facilitar a l'usuari que utilitzi les dues aplicacions l'aprenentatge i l'ús. Un altre punt a favor de la reutilització de components de *RPToolManager* per l'eina *RPPProjectManager* és que si ambdues aplicacions tenen una implementació i funcionament similar, el manteniment de les dues serà més fàcil.

16.2. Reutilització dels components per la finestra principal

La finestra principal, que està formada per diverses classes, prové d'una reutilització realitzada ja en la primera versió de l'aplicació *RPToolManager*. Els components provenen d'un treball de beca del grup de recerca GESSI per a les eines de selecció de components des-COTS QM, desenvolupat per Xavi López.

Les classes reutilitzades que implementen la visualització de la finestra principal són les que es descriuen a continuació:

- `GenericMainFrame`
- `InternalFrameGeneric`
- `GlobalValues`
- `SliderAccessDB`

16.2.1. GenericMainFrame

Aquesta classe va ser desenvolupada en el seu moment per Xavi López i reutilitzada a l'aplicació *RPToolManager* a partir de la versió 1. La classe `GenericMainFrame` és una classe de tipus `JFrame` que implementa una pantalla principal que mostra dues parts principals. Una part a la dreta en la que es pot mostrar una finestra i a l'esquerra una àrea on es poden anar afegint finestres.

La classe consta de mètodes per afegir els components a les dues bandes i eliminar el de la dreta, així com per afegir finestres noves per damunt de les dues parts. També hi ha mètodes que permeten afegir barres de menús i de tasques a la part de dalt de la finestra principal.

Pel que fa a la implementació, aquesta classe gestiona la mida de la finestra i els seus components mitjançant `Layouts` (l'usuari pot canviar la mida de la finestra al seu gust). La mida per defecte que s'assigna a la finestra és la que ocupa tota la pantalla.

16.2.2. InternalFrameGeneric

Al igual que la classe anterior, la classe *InternalFrameGeneric* va ser desenvolupada en el seu moment per Xavi López i reutilitzada en la primera versió de l'aplicació *RPToolManager*. Representa una finestra (subclasse d'*InternalFrame*) en la que es poden afegir tres components :

- Una barra de tasques (a la part superior de la finestra)
- Un arbre (a la part central de la finestra)
- Un panell que permet mostrar text en HTML (a la part inferior de la finestra).

Aquesta classe està orientada a facilitar la programació de finestres que mostren un arbre, obren un menú desplegable en prémer el botó dret sobre algun dels nodes i mostren certa informació al panell de text HTML quan es selecciona algun node de l'arbre.

Les finestres d'aquesta classe permeten que l'usuari les redimensioni, fins a una determinada mida mínima/màxima (si es vol canviar la que s'assigna per defecte, només cal cridar el mètode “*setMinimumSize*”/“*setMaximumSize*” des d'una subclasse un cop inicialitzada la finestra).

El panell de text i l'arbre es mostren dins de panells scroll, de manera que quan algun dels components es fa massa gran o la finestra es fa massa petita, apareixen barres d'scroll horitzontal o vertical. En el cas del panell de text només apareix barra d'scroll vertical.

Per reutilitzar aquesta classe cal implementar mètodes abstractes per:

- Inicialitzar la barra de tasques.
- Obtenir el node arrel de l'arbre ja inicialitzat (és a dir, amb tots els fills ja afegits).
- Mostrar algun menú desplegable quan es prem el botó dret del ratolí sobre algun dels nodes de l'arbre depenent de certes condicions.
- Inicialitzar els menús de botó dret (assignar oients, textos, etc...).

A més a més, el mètode “*internalFrameClosing*” queda pendent de definició per permetre a les subclasses fer determinades accions quan una finestra es tanca. La resta de mètodes que requereix *internalFrameListener* estan implementats però no fan res. Es poden sobreescriure si cal.

16.2.3. GlobalValues

Aquesta classe va ser desenvolupada en el seu moment per Gemma Grau i reutilitzada a l'aplicació *RPToolManager* des de la seva primera versió. Representa una Interface que conté les variables globals del sistema i que s'utilitzen en més d'un package. Conté, entre altres, les variables globals per als colors a utilitzar.

16.2.4. SliderAccessDB

La classe SliderAccessDB va ser desenvolupada en el seu moment per Xavi López i reutilitzada ja en la primera versió de l'aplicació *RPToolManager*. Es tracta d'una classe de tipus JDialog que implementa una vista que mostra una successió d'imatges per tal de mostrar a l'usuari que l'aplicació està buscant a la Base de Dades. També es mostra una JProgressBar en mode indefinit ja que no es té una mesura del temps que pot arribar a trigar la consulta.

Per tal d'utilitzar aquesta classe, només és necessari crear una nova instància d'ella amb la constructora "SliderAccessDB(GenericMainFrame owner)", inicialitzar-la amb el mètode "jbInit" i finalment establir el nom de la tasca que volem que aparegui a la finestra amb el mètode "setTaskName".

16.3. Reutilització dels components per a la visualització dels arbres

El component que permet visualitzar l'arbre, que està format per diverses classes, prové tant de la reutilització ja realitzada en la primera versió de l'aplicació *RPToolManager* com de les noves classes que es van crear per aquesta. Els components reutilitzats en la primera versió de *RPToolManager* provenen d'un treball de beca del grup de recerca GESSI per a les eines de selecció de components des-COTS QM, desenvolupat per Xavi López.

Les classes reutilitzades que implementen la visualització dels arbres són les que es descriuen a continuació:

- GenericTreeCellRenderer
- SortedMutableTreeNode
- GenericComparator

16.3.1. GenericTreeCellRenderer

Aquesta classe va ser desenvolupada en el seu moment per Xavi López i reutilitzada a l'aplicació *RPToolManager* a partir de la versió 1. Aquesta classe abstracta proveeix les funcionalitats per mostrar les entrades d'un arbre (un JTree) usant diferents icones. També permet mostrar un nombre determinat d'icones (o etiquetes) a continuació (a la dreta) de l'entrada principal de l'arbre.

Per fer tot això s'implementa el mètode "getTreeCellRendererComponent". Aquest mètode crida al mètode abstracte "determineIconsFromNode". Per reutilitzar la classe només cal heretar d'ella i implementar "determineIconsFromNode". Aquest mètode rep el Node de l'arbre (DefaultMutableTreeNode) a representar, i retorna un Vector amb les icones que es mostraran. La informació necessària caldrà extreure-la del "userObject" associat al DefaultMutableTreeNode en qüestió (que en general serà d'una classe pròpia

que contindrà precisament aquesta informació, introduïda en el moment de crear el `DefaultMutableTreeNode`).

Aquesta classe estén `TreePanel` ja que és la classe del Component retornat en “`getTreeCellRendererComponent`” (aquest mètode s’invoca automàticament sempre que cal obtenir una representació d’un Component que s’afegirà a l’arbre). El Component retornat consisteix en un `JPanel` on es van afegint diferents `JLabel` per representar les icones. La classe `TreePanel` facilita a “`getTreeCellRendererComponent`” la creació i posicionament dels components que s’afegiran. La constructora disposa d’un paràmetre que representa la distància que es posarà entre cada component afegit al Panel.

16.3.2. SortedMutableTreeNode

La classe `SortedMutableTreeNode` va ser desenvolupada per l’Oriol Arcas durant el desenvolupament de la primera versió de l’aplicació *RPToolManager*. Aquesta classe estén a `DefaultMutableTreeNode` i té per objectiu mantenir les entrades del `DefaultMutableTreeNode` ordenades segons el comparador per defecte `GenericComparator` (per més informació sobre aquest veure el següent apartat 16.3.3) També es pot ordenar amb un comparador propi utilitzant per crear el `SortedMutableTreeNode` la constructora “`SortedMutableTreeNode(Comparator comp)`”.

16.3.3. GenericComparator

Aquesta classe, a l’igual que l’anterior, va ser desenvolupada per Oriol Arcas durant el desenvolupament de la primera versió de l’aplicació *RPToolManager*. `GenericComparator` estén a `Comparator` i té per objectiu comparar dos objectes segons la seva conversió a string (realitzada amb el mètode “`toString`”).

16.4. Reutilització dels components per la finestra d’edició de projectes

La finestra d’edició de projectes, que està formada per diverses classes, prové de la reutilització de classes que es van crear per la primera versió de *RPToolManager*, desenvolupada per l’Oriol Arcas i l’Oriol Colléll.

Les classes reutilitzades que implementen la finestra d’edició de projectes són les que es descriuen a continuació:

- `OkCancelDialog`
- `GenericObjectTabs`
- `DocumentWithLimitCharacters`

16.4.1. OkCancelDialog

Aquesta classe va ser desenvolupada per Oriol Arcas durant el desenvolupament de la primera versió de l'aplicació *RPToolManager*. *OkCancelDialog*, que és una classe abstracta de tipus *JDialog*, té el propòsit de que s'implementi aquesta classe quan es vol tenir una finestra que realitza operacions sobre un objecte (en el nostre cas un projecte) que pot ser nou o no (la implementació del mètode abstracte “isNew” és qui ho ha de dir) i que aquestes operacions poden o no tenir èxit (la implementació del mètode abstracte “success” ha de retornar això precisament).

16.4.2. GenericObjectTabs

GenericObjectTabs, a l'igual que la classe anterior, va ser desenvolupada per Oriol Arcas durant el desenvolupament de la primera versió de l'aplicació *RPToolManager*. Aquesta classe, que és una classe de tipus *JTabbedPane*, implementa un panell amb pestanyes en les que apareixen, en el nostre cas, els camps per editar els atributs d'un projecte. A un *GenericObjectTabs* se li poden afegir més pestanyes de les que ja té amb el mètode “addNewPanel”.

16.4.3. DocumentWithLimitCharacters

Aquesta classe, que estén de *PlainDocument*, va ser realitzada per mi durant el desenvolupament de la 2^a versió de *RPToolManager*. El propòsit d'aquesta classe és limitar el nombre de caràcters que es poden inserir en un *JTextField* mentre aquest s'està editant. Per aconseguir-ho, el mètode “insertString” es crida automàticament cada cop que s'intenta insertar caràcters a l'editor i comprova que no es sobrepassi el nombre màxim de caràcters establert; si el nombre màxim no es sobrepassa, es crida al mètode de la classe pare per a que s'insertin els caràcters; si es sobrepassa el límit, el mètode retorna sense fer res. En l'aplicació *RPPProjectManager* utilitzarem aquesta classe per tal de limitar el nombre de caràcters que pot tenir el nom d'un projecte o d'un requisit, entre d'altres.

17. Base de Dades

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager. 79	
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
17.1. Mapeig per Hibernate	196
17.2. Consideracions sobre el mapeig.....	198
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

17.1. Mapeig per Hibernate

Per tal de fer funcionar Hibernate, entre d'altres, fa falta un fitxer .hbm, més conegut com fitxer de mapeig, que permet al programador definir com és el seu model de dades, quines relacions existeixen i quina forma tenen.

A continuació es mostra quin és el contingut del fitxer de mapeig per la 1^a versió de l'aplicació *RPPProjectManager*.

```
<hibernate-mapping>
```

```
<class name="edu.upc.gessi.rpproject.domain.project.Project"
table="PROJECT">

  <id name="id" column="ID" type="long">
    <generator class="native"/>
  </id>

  <property name="name" column="NAME" type="string" not-null="true"
unique="true"/>
  <property name="description" column="DESCRIPTION" type="text"
long="2000" not-null="false"/>
  <property name="creationDate" type="timestamp"
column="CREATION_DATE" not-null="true"/>
  <property name="modificationDate" type="timestamp"
column="MODIFICATION_DATE" not-null="true"/>

  <set name="myRequirements" lazy="false">
    <key column="MY_PROJECT"/>
    <one-to-many class="edu.upc.gessi.rpproject.domain.requirement.
Requirement"/>
  </set>

</class>

<class name="edu.upc.gessi.rpproject.domain.requirement.Requirement"
table="REQUIREMENT">

  <id name="id" column="ID" type="long">
    <generator class="native"/>
  </id>

  <property name="name" column="NAME" type="string" not-null="true"/>
  <property name="questionText" column="QUESTION_TEXT" type="text"
length="2000" not-null="true"/>
  <property name="formText" column="PATTERN_TEXT" type="text"
length="2000" not-null="true"/>

  <joined-subclass name="edu.upc.gessi.rpproject.domain.requirement.
PatternInstanceRequirement" table="REQUIREMENT_PATTERN_INSTANCE">

    <key column="ID"/>

    <property name="idMyRequirementPattern" column="ID_MY_REQ_
PATTERN" type="long" not-null="true"/>
```

```

        <property name="idMyRequirementForm" column="ID_MY_REQ_FORM"
        type="long" not-null="true"/>
        <property name="idMyPatternItem" column="ID_MY_PATTERN_ITEM"
        type="long" not-null="true"/>

        <set name="values" lazy="false">
            <key column="MY_INSTANCE"/>
            <one-to-many class="edu.upc.gessi.rpproject.domain.value.
            Value"/>
        </set>

    </joined-subclass>

    <joined-subclass name="edu.upc.gessi.rpproject.domain.requirement.
    NoPatternInstanceRequirement" table="NO_REQUIREMENT_PATTERN_INSTANCE">

        <key column="ID"/>

        <joined-subclass name="edu.upc.gessi.rpproject.domain.
        requirement.RegWithPatternAssociated" table=
        "REQ_WITH_PATTERN_ASSOCIATED">
            <key column="ID"/>
            <property name="idMyRequirementPattern" column="ID_MY_REQ_
            PATTERN" type="long" not-null="true"/>
            <property name="idMyRequirementForm" column="ID_MY_PATTERN
            _ITEM" type="long" not-null="true"/>
        </joined-subclass>

    </joined-subclass>

</class>

<class name="edu.upc.gessi.rpproject.domain.value.Value" table="VALUE">

    <id name="id" column="ID" type="long">
        <generator class="native"/>
    </id>

    <property name="idMyParameter" column="ID_MY_PARAMETER" type="long"
    not-null="false"/>

    <joined-subclass name="edu.upc.gessi.rpproject.domain.value.
    SimpleValue" table="SIMPLE_VALUE">

        <key column="ID"/>
        <property name="satisfyRestriction" column="SATISFY_RESTRICTION"
        type="boolean" not-null="true"/>
        <property name="defined" column="IS_DEFINED" type="boolean" not-
        null="true"/>

        <joined-subclass name="edu.upc.gessi.rpproject.domain.value.
        FloatValue" table="FLOAT_VALUE">
            <key column="ID"/>
            <property name="value" column="VALUE" type="float" not-null="true"/>
        </joined-subclass>

        <joined-subclass name="edu.upc.gessi.rpproject.domain.value.
        IntegerValue" table="INTEGER_VALUE">
            <key column="ID"/>
            <property name="value" column="VALUE" type="int" not-null="true"/>
        </joined-subclass>

```

```

<joined-subclass name="edu.upc.gessi.rpproject.domain.value.
StringValue" table="STRING_VALUE">
  <key column="ID"/>
  <property name="value" column="VALUE" type="string" not-
    null="true"/>
</joined-subclass>

<joined-subclass name="edu.upc.gessi.rpproject.domain.value.
TimePointValue" table="TIME_POINT_VALUE">
  <key column="ID"/>
  <property name="year" column="YEAR_1" type="int" not-
    null="false"/>
  <property name="month" column="MONTH_1" type="int" not-
    null="false"/>
  <property name="day" column="DAY_1" type="int" not-
    null="false"/>
  <property name="hour" column="HOURL_1" type="int" not-
    null="false"/>
  <property name="minute" column="MINUTE_1" type="int" not-
    null="false"/>
  <property name="second" column="SECOND_1" type="int" not-
    null="false"/>
</joined-subclass>

<joined-subclass name="edu.upc.gessi.rpproject.domain.value.
DomainValue" table="DOMAIN_VALUE">
  <key column="ID"/>
  <property name="value" column="VALUE" type="string" not-
    null="true"/>
  <property name="idMyDomain" column="ID_MY_DOMAIN" type="long"
    not-null="false"/>
</joined-subclass>

</joined-subclass>

<joined-subclass name="edu.upc.gessi.rpproject.domain.value.
SetValue" table="SET_VALUE">
  <key column="ID"/>
  <set name="myValues" lazy="false">
    <key column="MY_SET"/>
    <one-to-many class="edu.upc.gessi.rpproject.domain.value.
      SimpleValue"/>
  </set>
</joined-subclass>

</class>

</hibernate-mapping>

```

17.2. Consideracions sobre el mapeig

Els atributs *id* de totes les classes que el tenen es generen de forma automàtica a la base de dades, despreocupant al programador d'aquesta tasca. Els atributs que són de tipus simple (enters, floats, strings, dates, etc) es mapegen d'una forma molt simple mitjançant el tag *property*. En quant als atributs que són pròpiament objectes definits pel programador o conjunt d'aquests, cal destacar:

- **Relacions one-to-many.** Les relacions one-to-many, que correspon a tenir un atribut que és un conjunt d'objectes dins d'una classe, es mapegen tal com segueix. Ho explicarem mitjançant un exemple. Un SetValue pot tenir zero o més SimpleValue associats, però un SimpleValue només pot pertànyer a un SetValue o a cap. Per tant, això és una relació one-to-many en la navegabilitat de SetValue a SimpleValue. El que farem és, doncs, al mapeig de la classe SetValue afegir:

```
<set name="myValues" lazy="false">
  <key column="MY_SET"/>
  <one-to-many class="edu.upc.gessi.rpproject.domain.value.
    SimpleValue"/>
</set>
```

Tenint en compte que l'atribut a la classe SetValue que és un conjunt de SimpleValue és *myValues*.

- **Relacions amb els objectes guardats a altres bases de dades (patterns, forms, parameters, etc.).** Degut a que els patrons i tots els seus elements estan guardats en una base de dades diferents d'on es guarden els projectes i els seus requisits, és impossible mapejar directament la relació d'un element que es guarda a la base de dades de projectes amb un element que es guarda a la base de dades dels patrons. Així doncs, per guardar les relacions que es poden veure al model conceptual entre els elements d'ambdues bases de dades (veure capítol 13 per més informació sobre el model conceptual), el que s'ha fet és, en comptes de mapejar l'objecte de la base de dades de patrons al mapeig de la base de dades de projectes, s'ha guardat com un atribut el seu identificador. Per exemple, un Value (base de dades de projectes) té una associació amb un Parameter (base de dades de patrons); així doncs, en comptes de mapejar a la classe Value explícitament una relació many-to-one amb la classe Parameter, el que hem fet és afegir a la classe Value un tag property on es guarda l'identificador del paràmetre amb el que està associat:

```
<property name="idMyParameter" column="ID_MY_PARAMETER"
  type="long" not-null="false"/>
```


18. Document de Disseny

1.	Introducció.....	11
2.	Patrons de requisits.....	15
3.	Eines de Treball.....	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades.....	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager. 79		
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús.....	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades.....	195
18.	Document de Disseny.....	201
18.1.	<i>Visualització de projectes i patrons.....</i>	<i>202</i>
18.1.1.	Visualitzar tots els projectes en ordre alfabètic.....	202
18.1.2.	Visualitzar els atributs d'un projecte.....	204
18.1.3.	Visualitzar tota la informació d'un projecte.....	205
18.1.4.	Visualitzar tots els patrons segons el classification schema seleccionat	205
18.1.5.	Visualitzar els atributs d'un element d'un patró.....	210
18.1.6.	Visualitzar tota la informació d'un patró	211
18.2.	<i>Construcció de projectes</i>	<i>212</i>
18.2.1.	Crear / Editar un projecte	212
18.2.2.	Crear / Editar un requisit "nou"	216
18.2.3.	Crear / Editar un requisit amb un patró associat.....	219
18.2.4.	Crear / Editar un requisit instància d'un patró	223
18.2.5.	Eliminar un requisit d'un projecte	229
18.2.6.	Modificar un valor d'un requisit	230
18.3.	<i>Gestió de projectes</i>	<i>232</i>
18.3.1.	Copiar un projecte	232
18.3.2.	Eliminar un projecte.....	234
18.4.	<i>Interoperabilitat</i>	<i>234</i>
18.4.1.	Generar pdf amb tota la informació d'un projecte	234
18.4.2.	Generar pdf amb el document call-for-tenders d'un projecte.....	237
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
21.	Future Work	251
22.	Conclusions.....	255
23.	Bibliografia	257
24.	Contingut del CD.....	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

18.1. Visualització de projectes i patrons

18.1.1. Visualitzar tots els projectes en ordre alfabètic

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries per poder mostrar la finestra amb l'arbre que conté els noms de tots els projectes que hi ha al repositori.

Nota 1: Al fer un new de projectsInternalFrame és crida a “jbInit”, que inicialitza tots els components de la finestra i crida a “loadProjectsAlphabetically”, que és qui s'encarrega de generar l'arbre que conté la llista de tots els projectes ordenats alfabèticament. Totes les crides que es realitzen després de “2a. loadProjectsAlphabetically” es realitzen des d'aquesta operació.

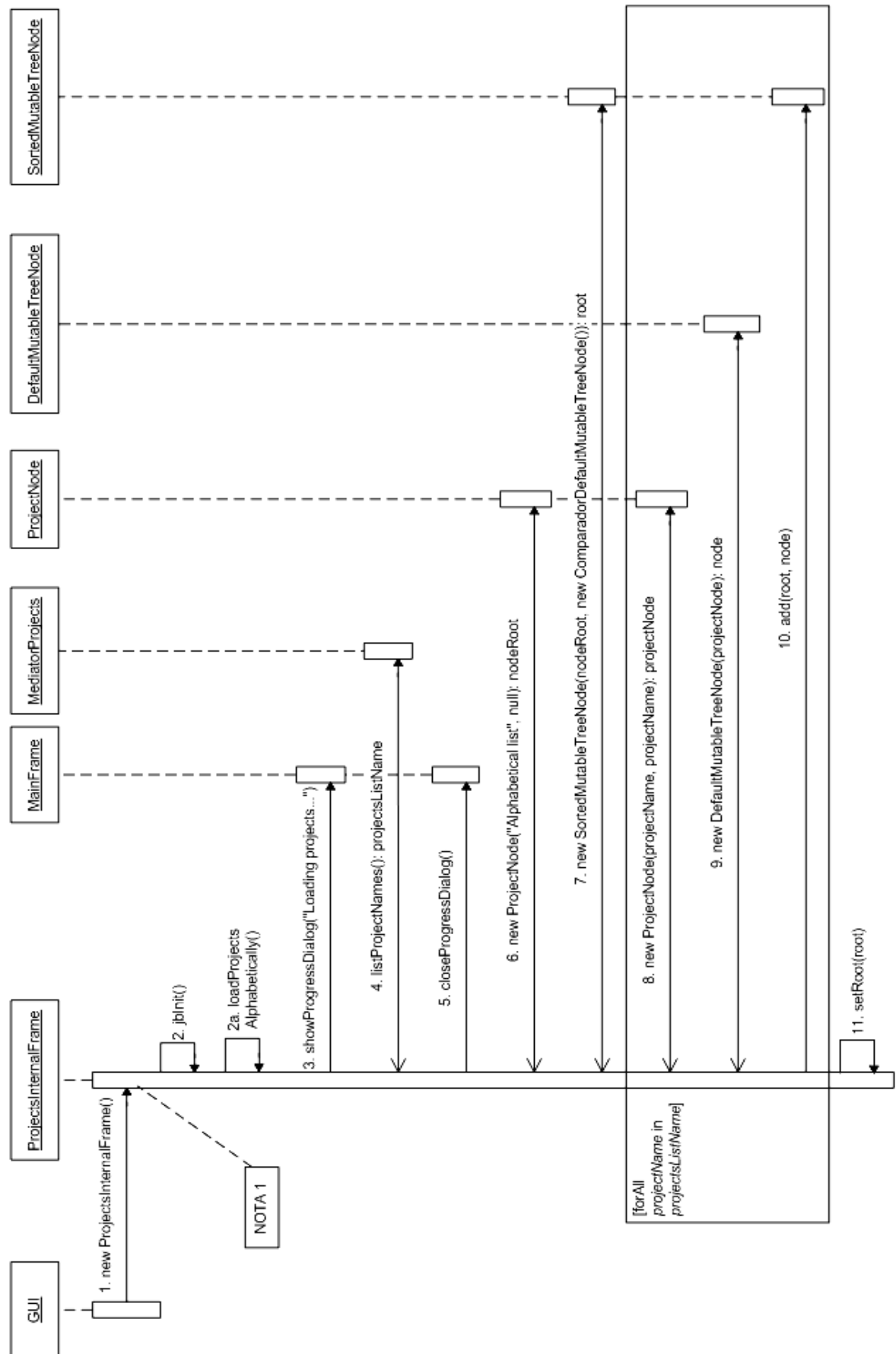


Figura 25: Diagrama de disseny de "Visualitzar tots els projectes en ordre alfabètic"

18.1.2. Visualitzar els atributs d'un projecte

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries per visualitzar a la finestra de projectes, a la part del fons taronja, els atributs del projecte seleccionat a l'arbre.

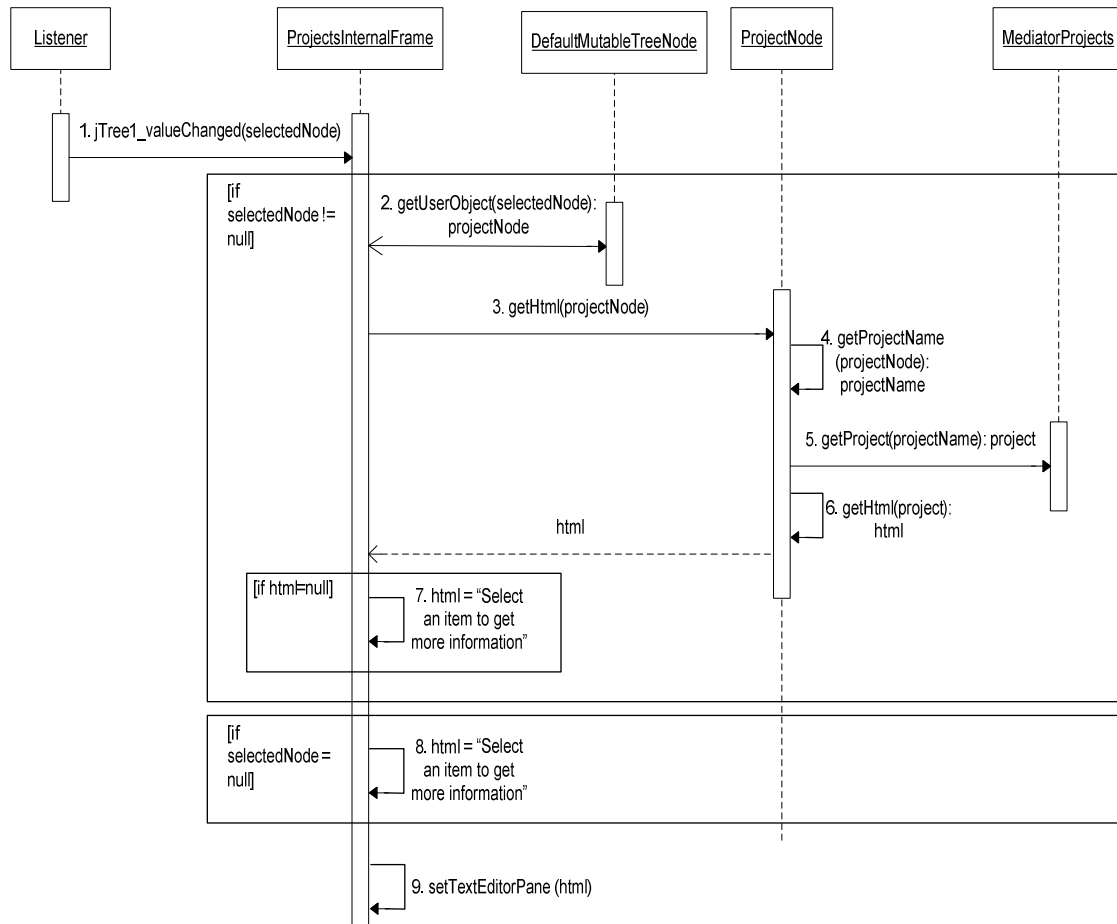


Figura 26: Diagrama de disseny de "Visualitzar els atributs d'un projecte"

18.1.3. Visualitzar tota la informació d'un projecte

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries per visualitzar en una nova finestra tota la informació del projecte seleccionat a l'arbre.

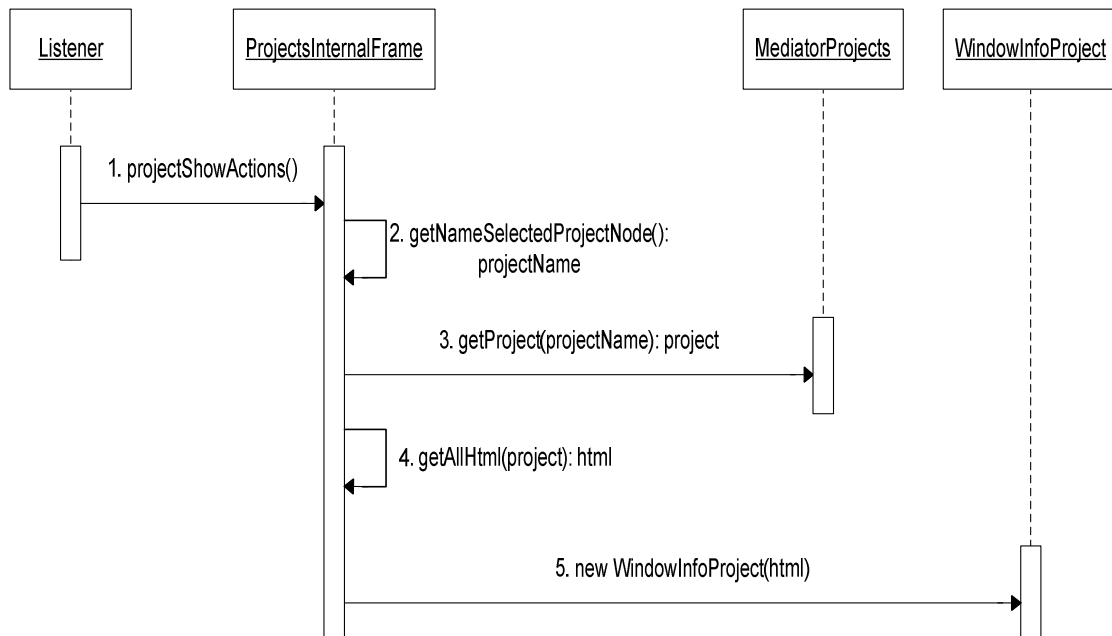


Figura 27: Diagrama de disseny de "Visualitzar tota la informació d'un projecte"

18.1.4. Visualitzar tots els patrons segons el classification schema seleccionat

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries per poder mostrar la finestra amb l'arbre que conté els noms de tots els patrons que hi ha al repositori junt amb els seus elements (forms, fixed i extended parts) segons el classification schema seleccionat al desplegable que conté la llista de tots els classification schema. Cal tenir en compte que només executem el codi corresponent al següent diagrama de disseny quan realment el classification schema ha canviat (si al desplegable amb tots els classification schema tornem a seleccionar el classification schema que hi havia carregat no fem res).

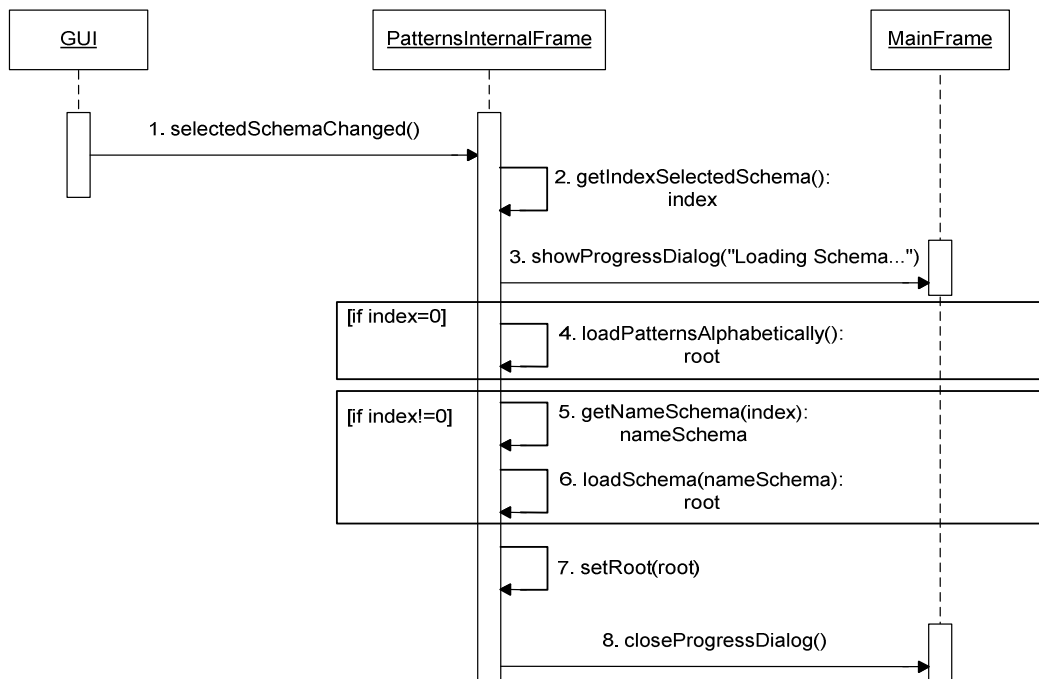


Figura 28: Diagrama de disseny 1 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

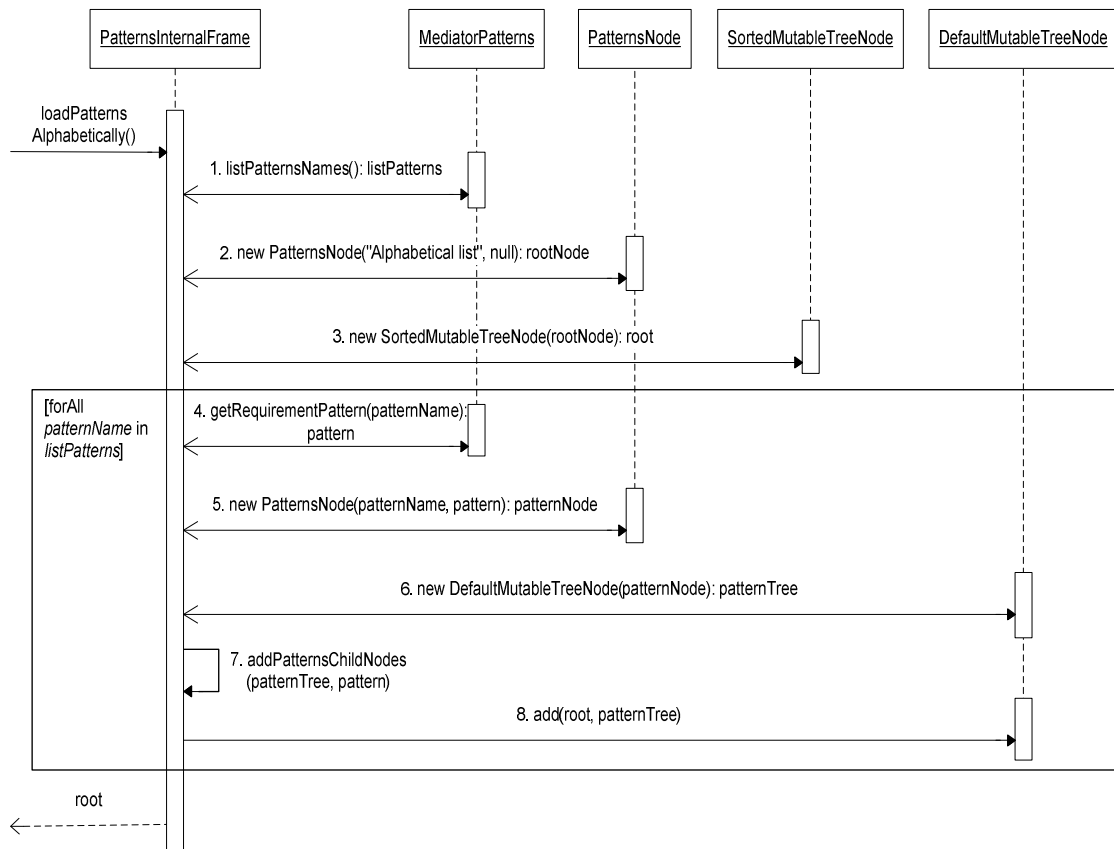


Figura 29: Diagrama de disseny 2 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

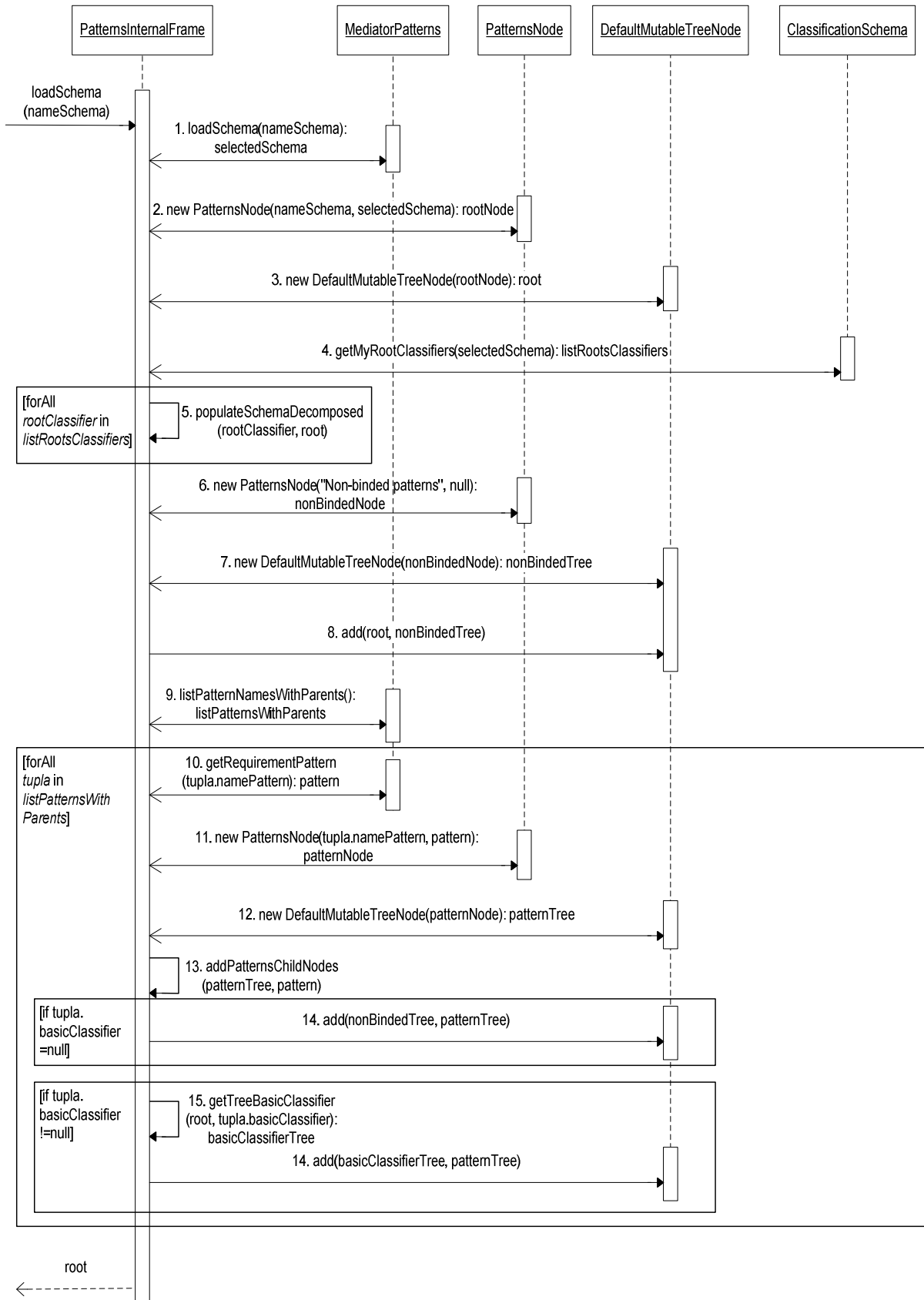


Figura 30: Diagrama de disseny 3 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

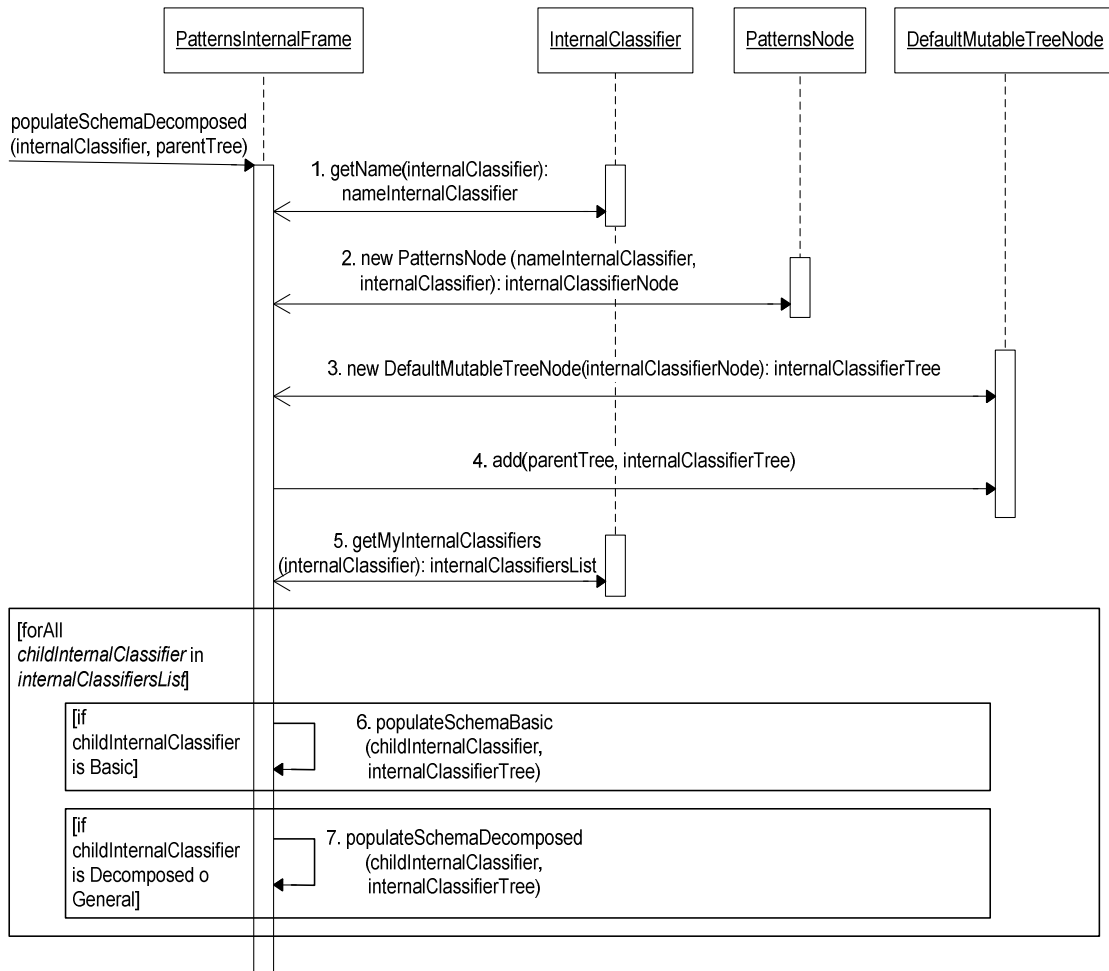


Figura 31: Diagrama de disseny 4 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

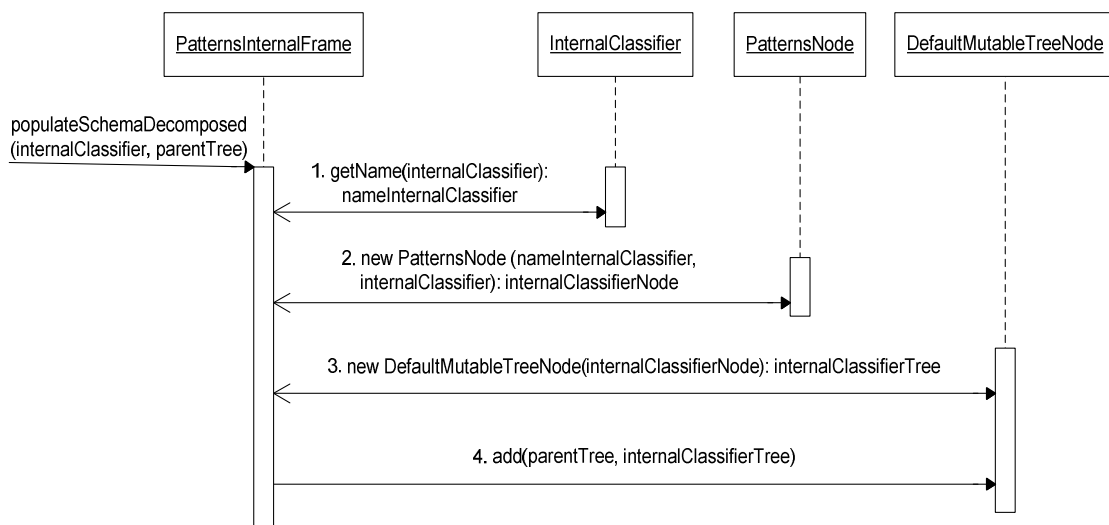


Figura 32: Diagrama de disseny 5 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

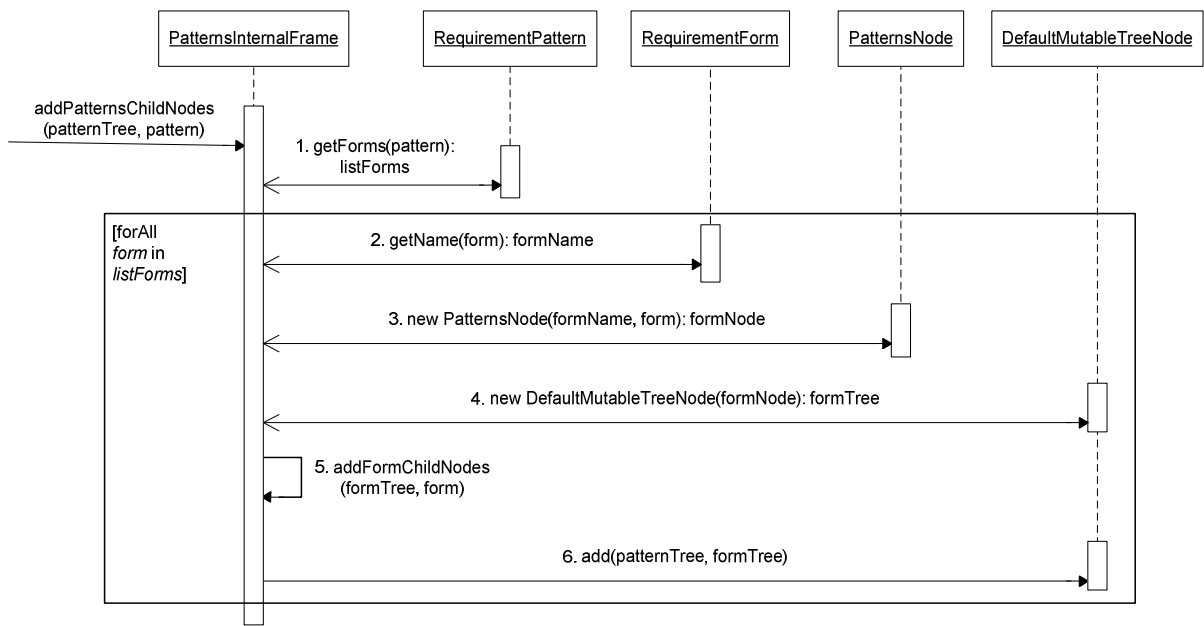


Figura 33: Diagrama de disseny 6 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

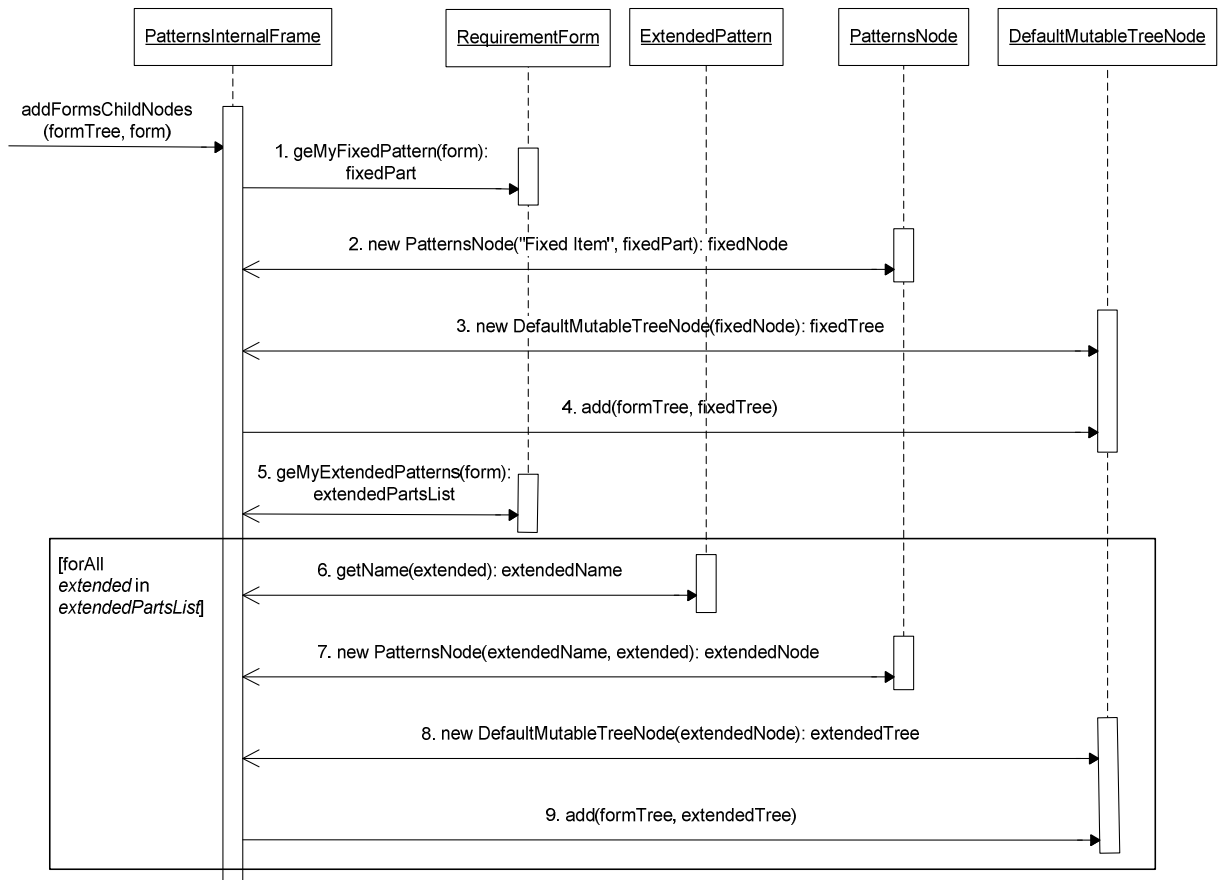


Figura 34: Diagrama de disseny 7 de 7 de "Visualitzar tots els patrons segons el classification schema seleccionat"

18.1.5. Visualitzar els atributs d'un element d'un patró

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries per visualitzar a la finestra de patrons, a la part del fons taronja, els atributs d'un dels elements d'un patró, el que hi ha seleccionat a l'arbre (que pot ser el patró en si mateix, o un dels seus forms, fixed o extended parts).

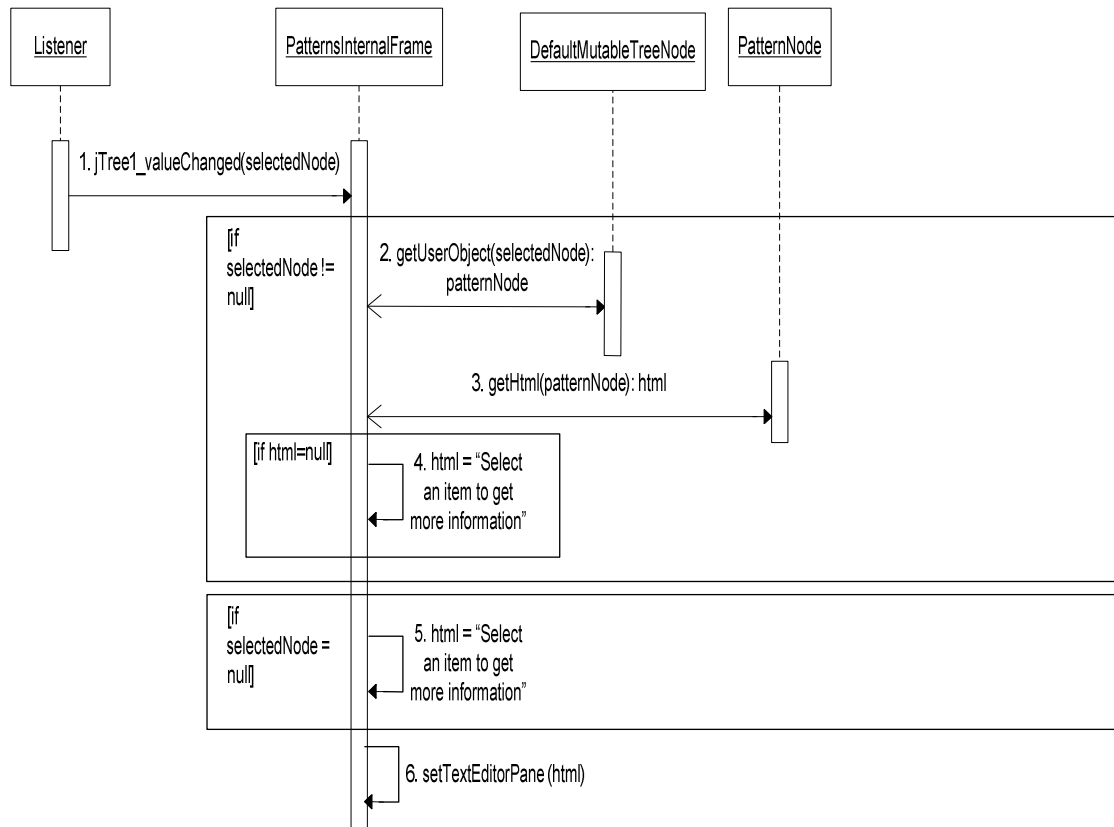


Figura 35: Diagrama de disseny de "Visualitzar els atributs d'un element d'un patró"

18.1.6. Visualitzar tota la informació d'un patró

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries per visualitzar en una nova finestra tota la informació del patró seleccionat a l'arbre. Cal tenir en compte que només mostrarem la finestra amb tota la informació del patró quan el node que es seleccioni correspongui a un patró; en altre cas no es farà res.

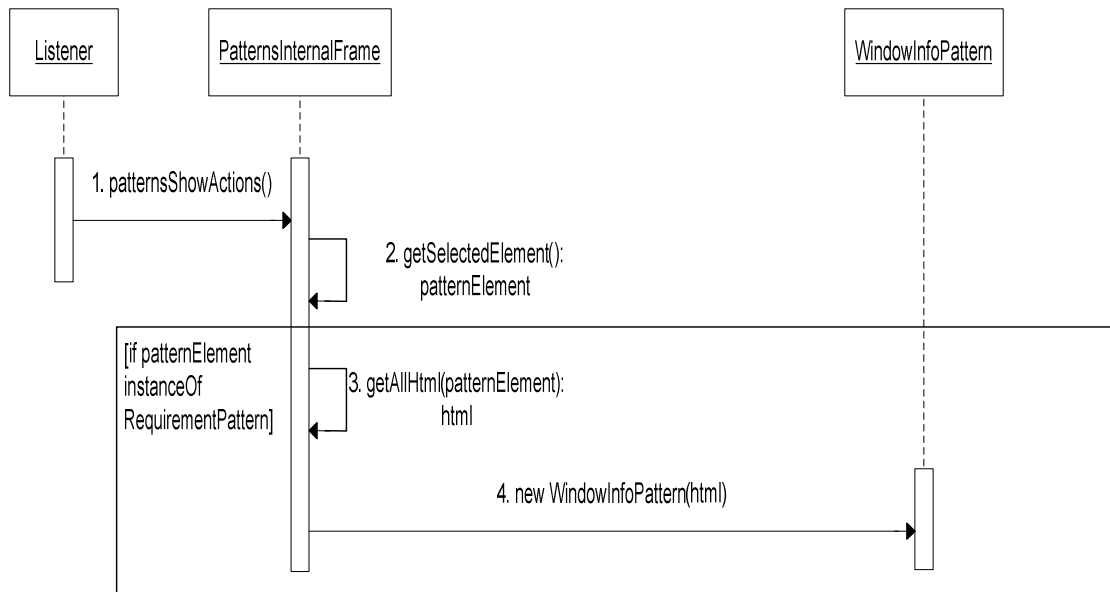


Figura 36: Diagrama de disseny de "Visualitzar tota la informació d'un patró"

18.2. Construcció de projectes

18.2.1. Crear / Editar un projecte

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten crear un nou projecte i guardar-lo al repositori de projectes o editar un projecte i guardar els seus canvis al repositori.

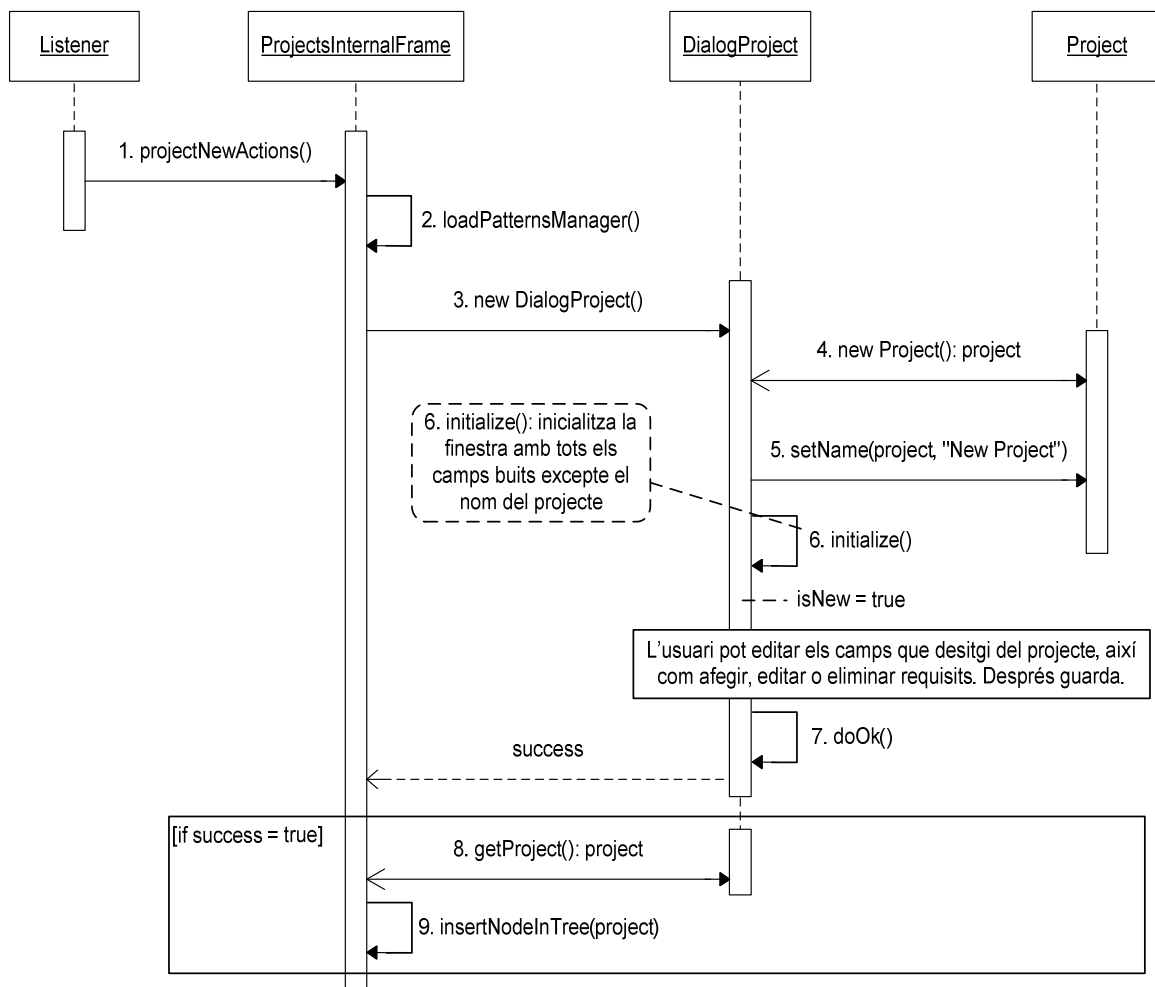


Figura 37: Diagrama de disseny 1 de 6 de "Crear / Editar un projecte"

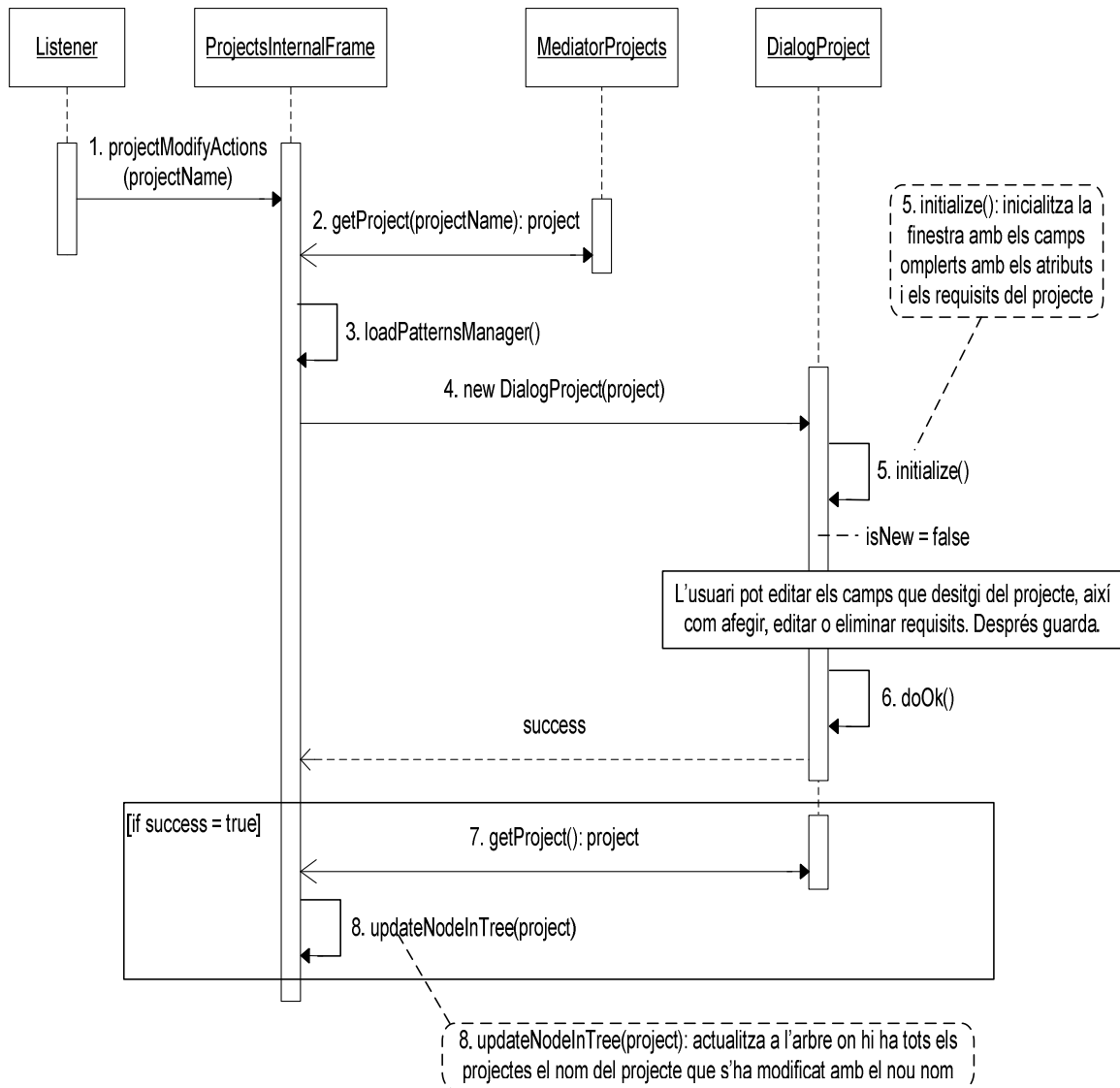


Figura 38: Diagrama de disseny 2 de 6 de "Crear / Editar un proyecto"

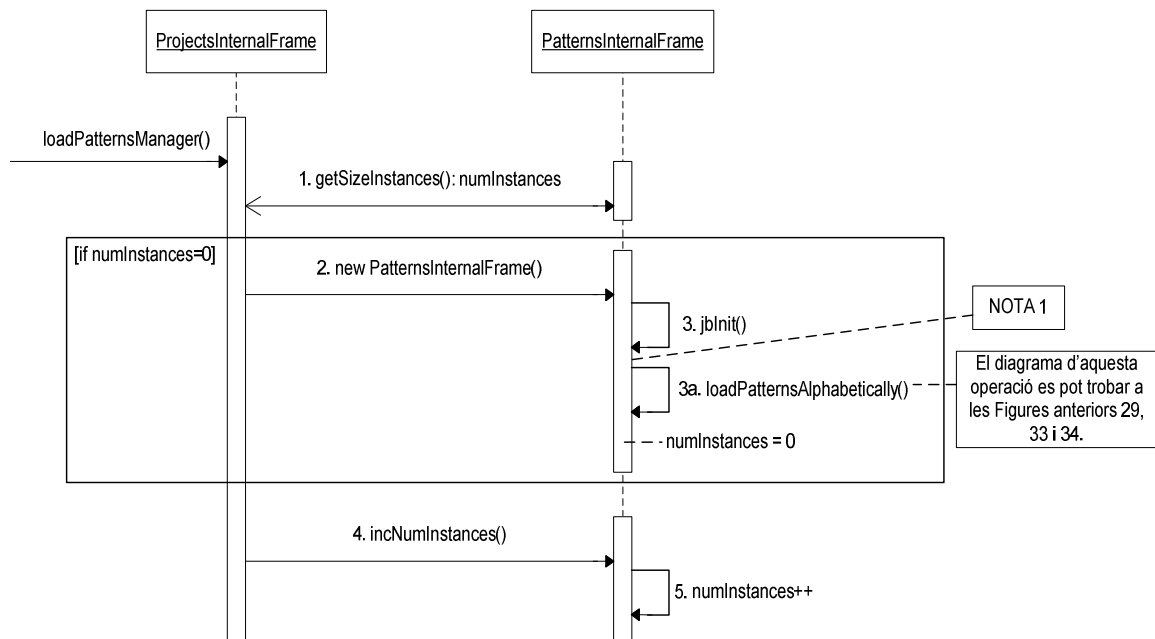


Figura 39: Diagrama de disseny 3 de 6 de "Crear / Editar un projecte"

Nota 1: Al fer un new de patternsInternalFrame és crida a "jblnit", que inicialitza tots els components de la finestra i crida a "loadPatternsAlphabetically", que és qui s'encarrega de generar l'arbre que conté la llista de tots els patrons ordenats alfabèticament.

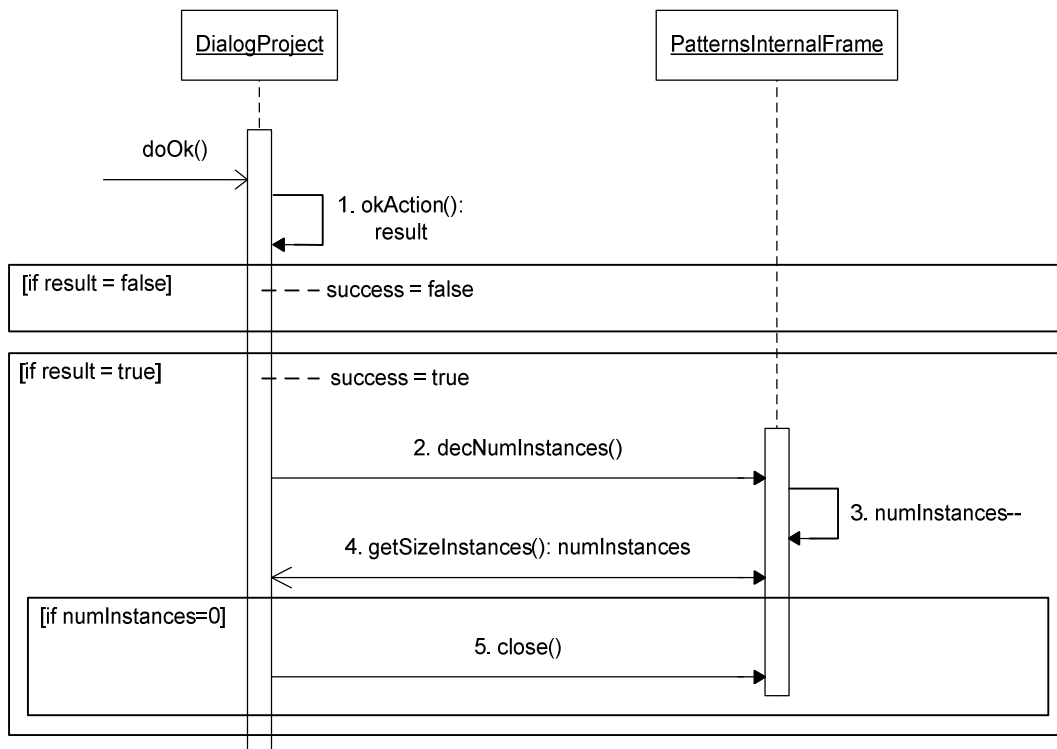


Figura 40: Diagrama de disseny 4 de 6 de "Crear / Editar un projecte"

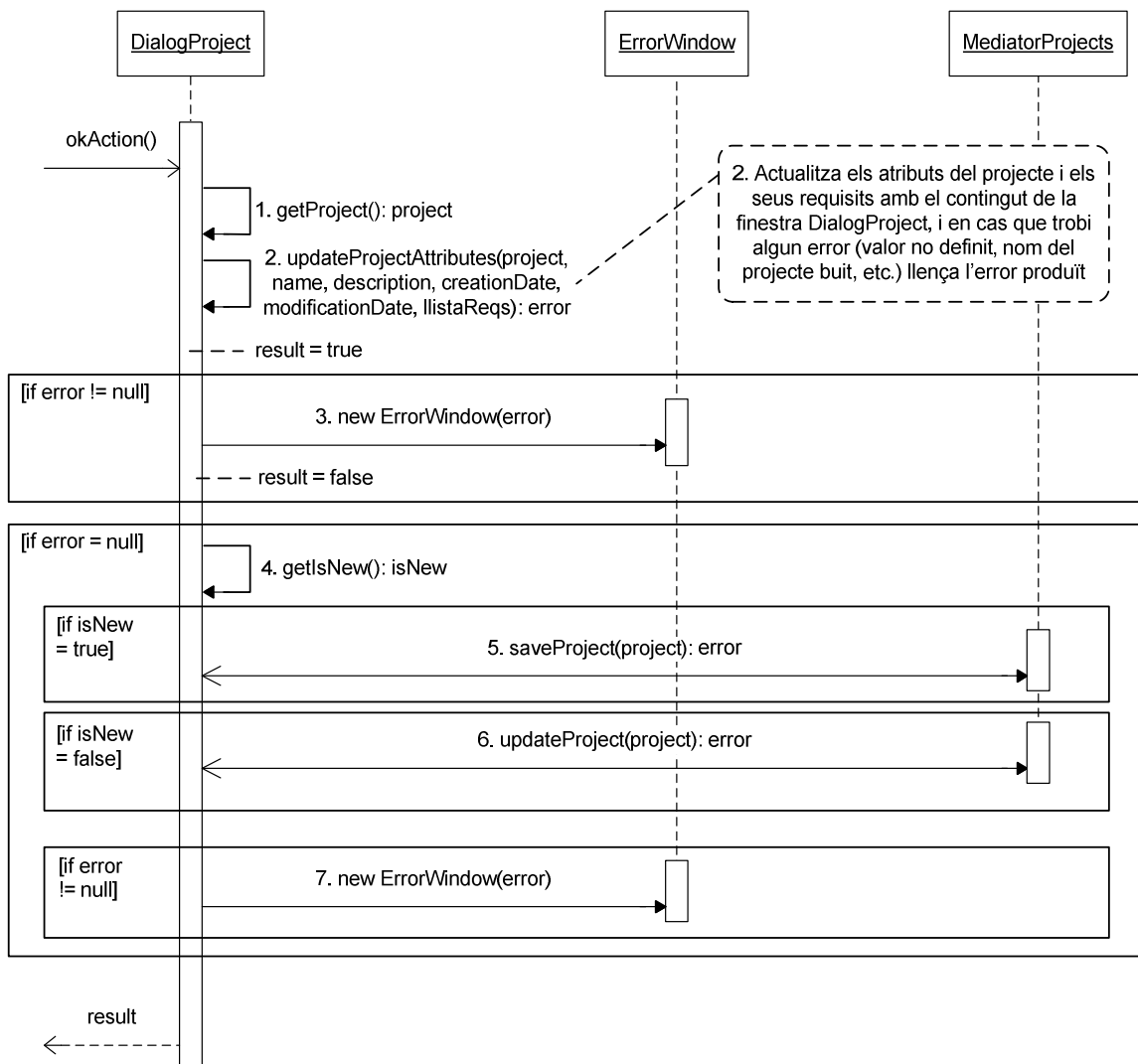


Figura 41: Diagrama de disseny 5 de 6 de "Crear / Editar un projecte"

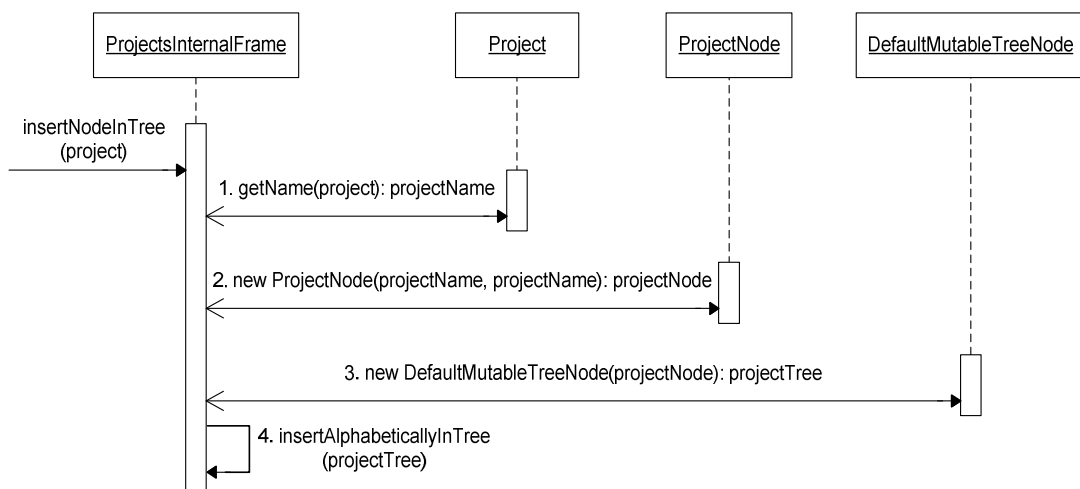


Figura 42: Diagrama de disseny 6 de 6 de "Crear / Editar un projecte"

18.2.2. Crear / Editar un requisit “nou”

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten crear un nou requisit de tipus “nou” dins d’un projecte o bé editar-lo.

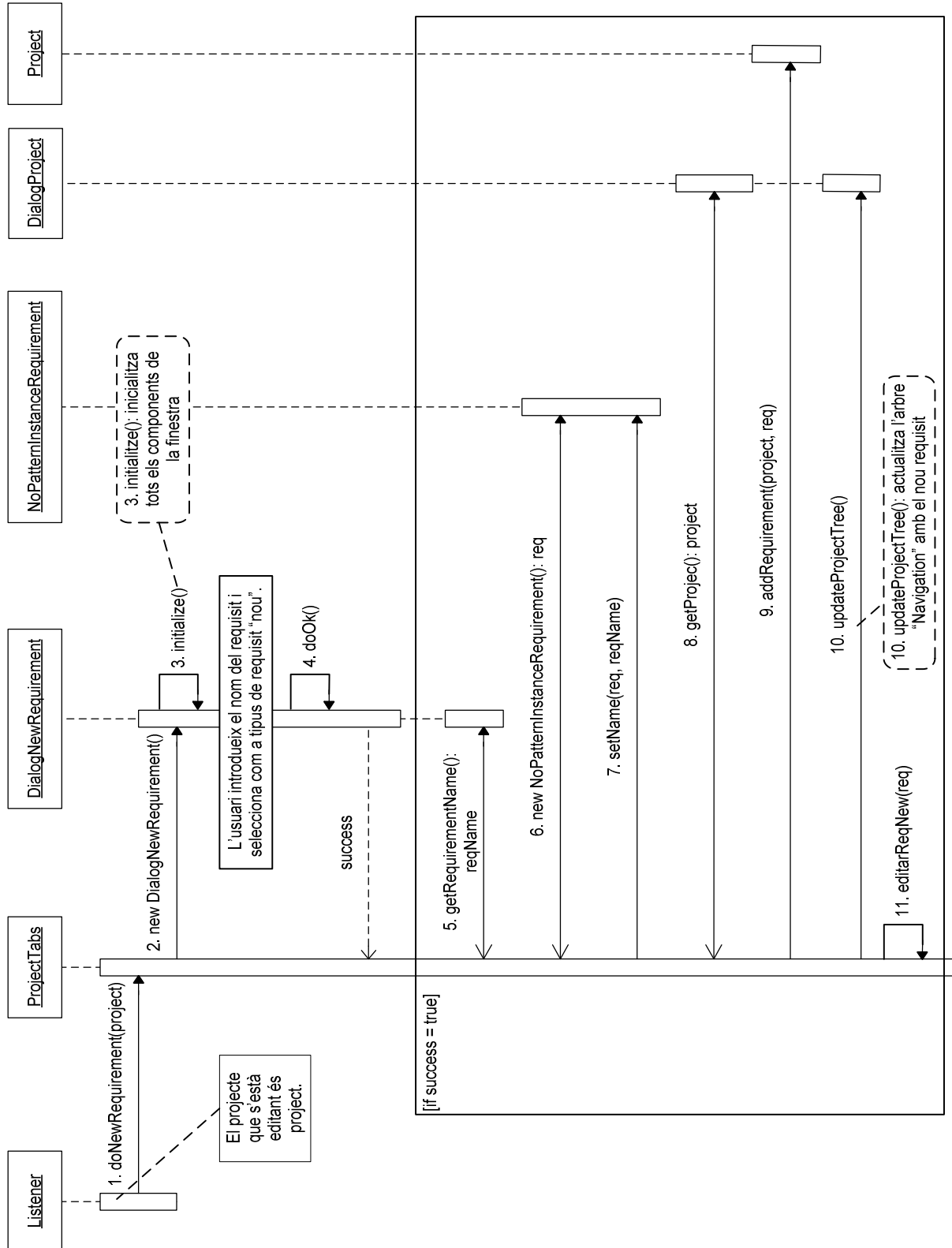


Figura 43: Diagrama de disseny 1 de 6 de "Crear / Editar un requisit 'nou' "

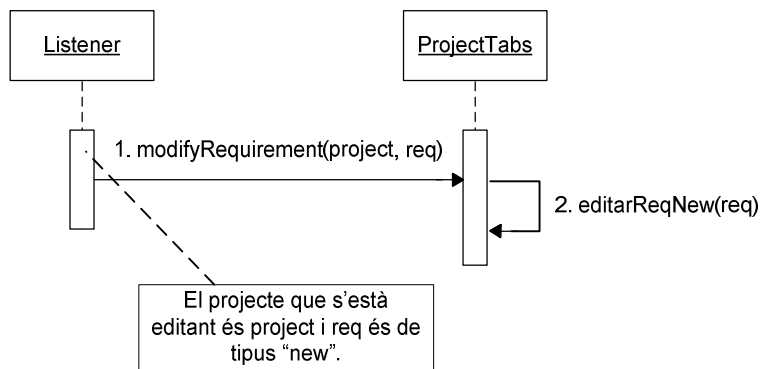


Figura 44: Diagrama de disseny 2 de 6 de "Crear / Editar un requisit 'nou' "

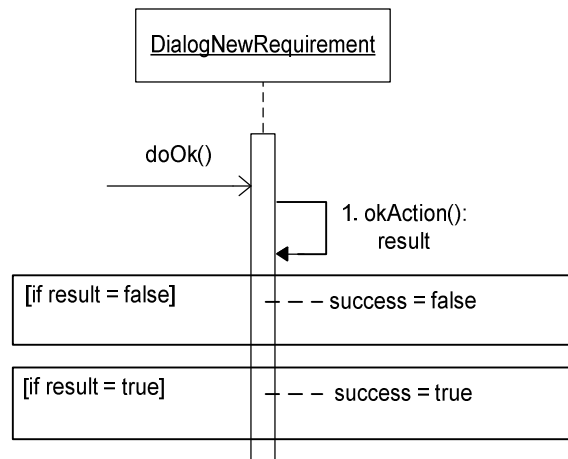


Figura 45: Diagrama de disseny 3 de 6 de "Crear / Editar un requisit 'nou' "

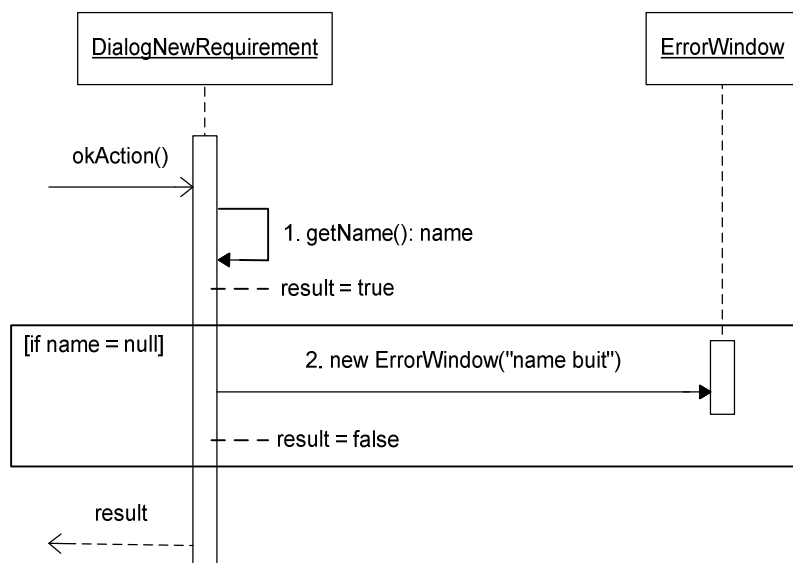


Figura 46: Diagrama de disseny 4 de 6 de "Crear / Editar un requisit 'nou' "

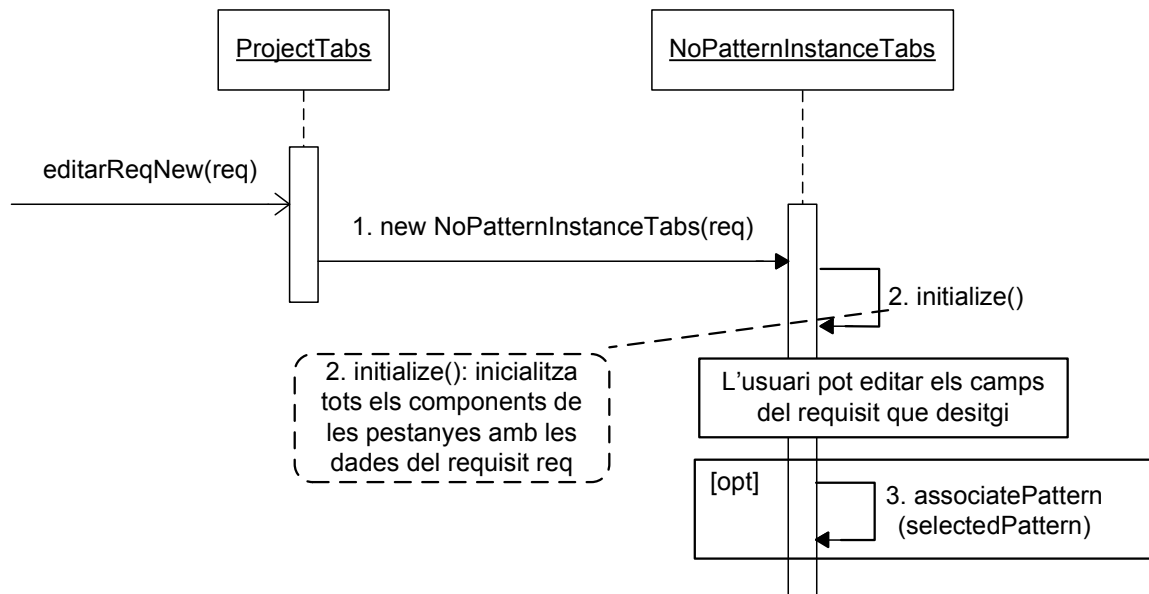


Figura 47: Diagrama de disseny 5 de 6 de "Crear / Editar un requisit 'nou' "

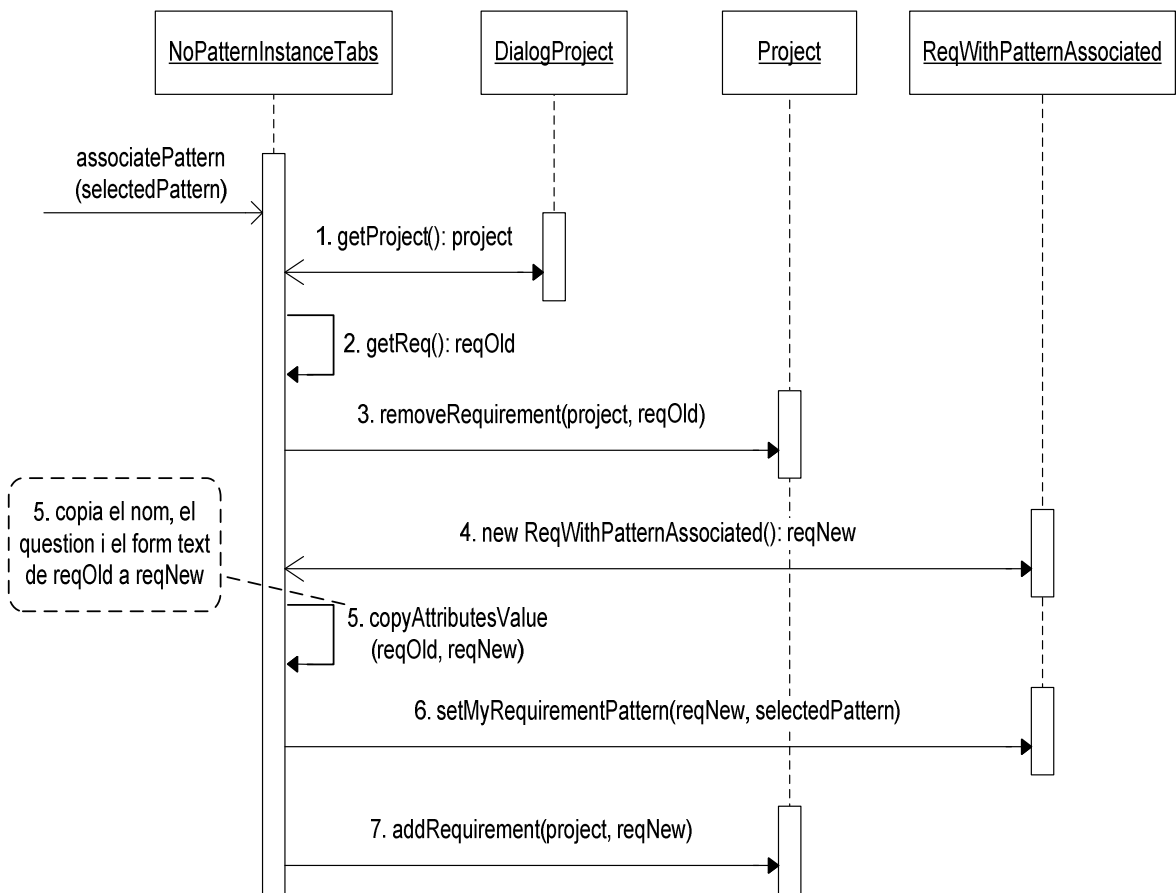


Figura 48: Diagrama de disseny 6 de 6 de "Crear / Editar un requisit 'nou' "

18.2.3. Crear / Editar un requisit amb un patró associat

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten crear un nou requisit de tipus “amb patró associat” dins d’un projecte o bé editar-lo.

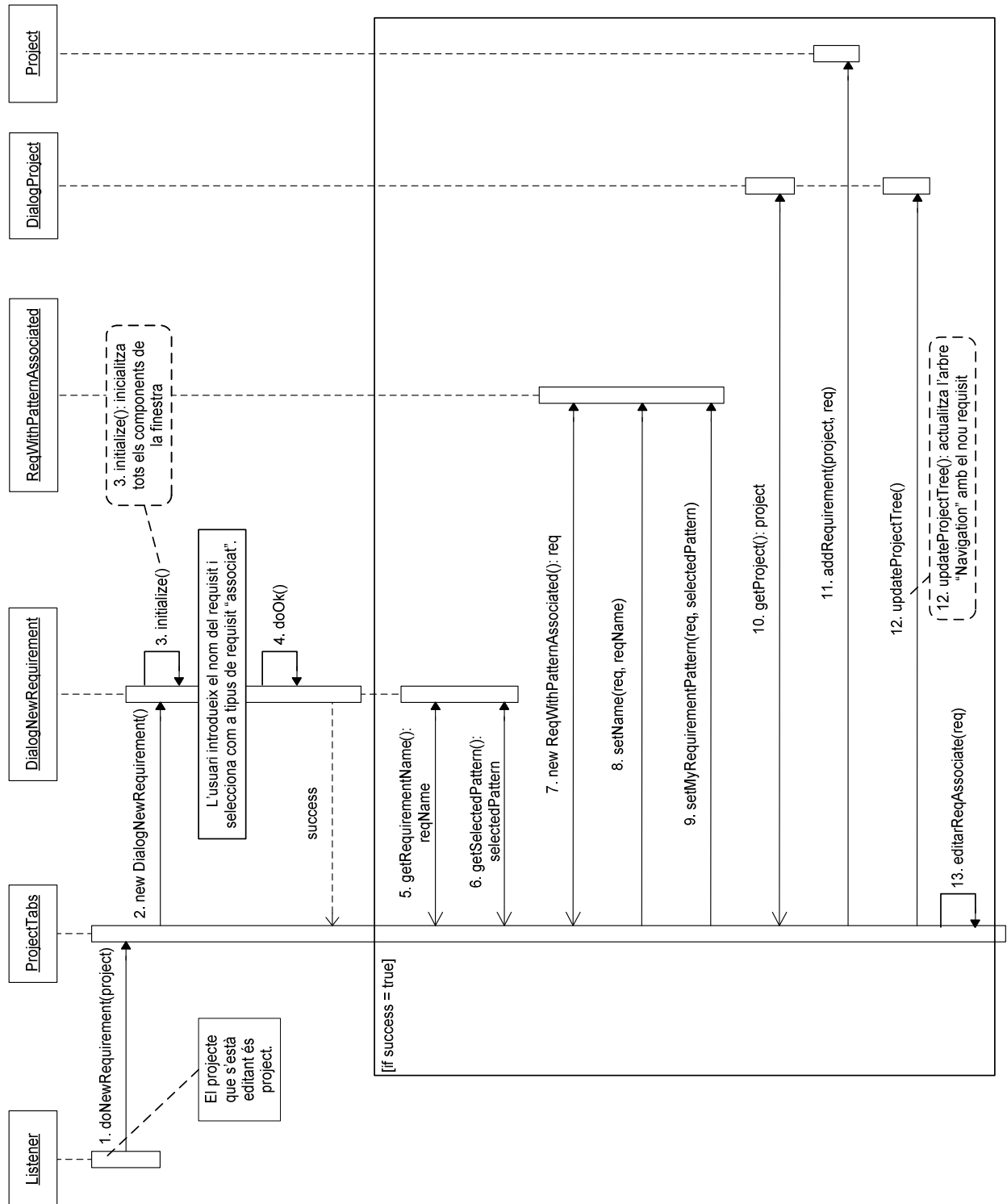


Figura 49: Diagrama de disseny 1 de 7 de "Crear / Editar un requisit amb patró associat"

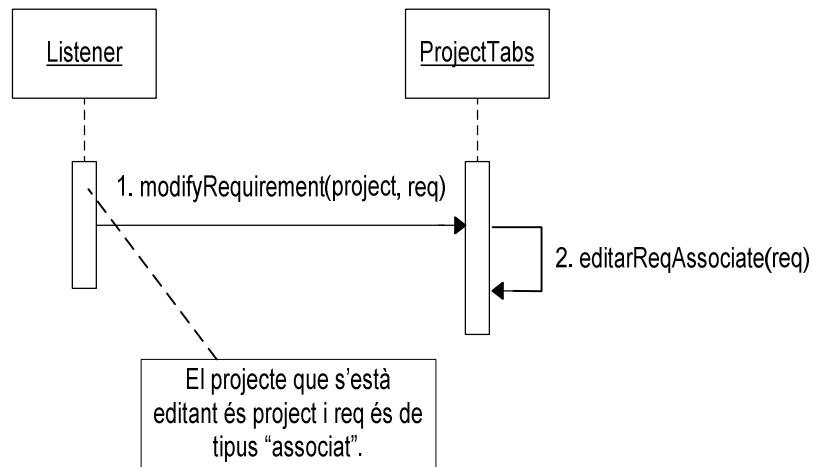


Figura 50: Diagrama de disseny 2 de 7 de "Crear / Editar un requisit amb patró associat"

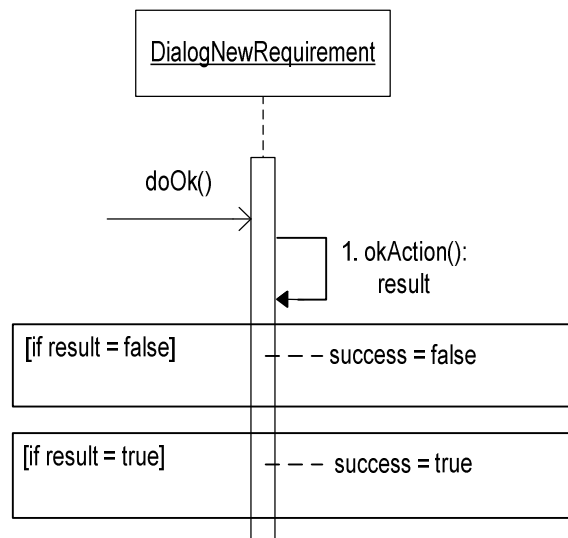


Figura 51: Diagrama de disseny 3 de 7 de "Crear / Editar un requisit amb patró associat"

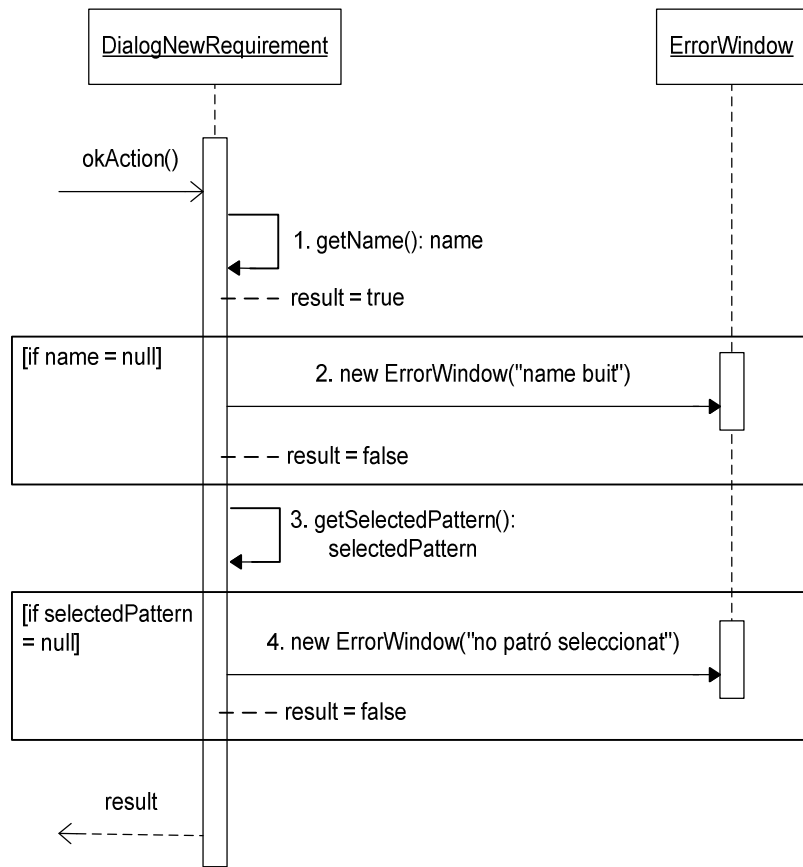


Figura 52: Diagrama de disseny 4 de 7 de "Crear / Editar un requisit amb patró associat"

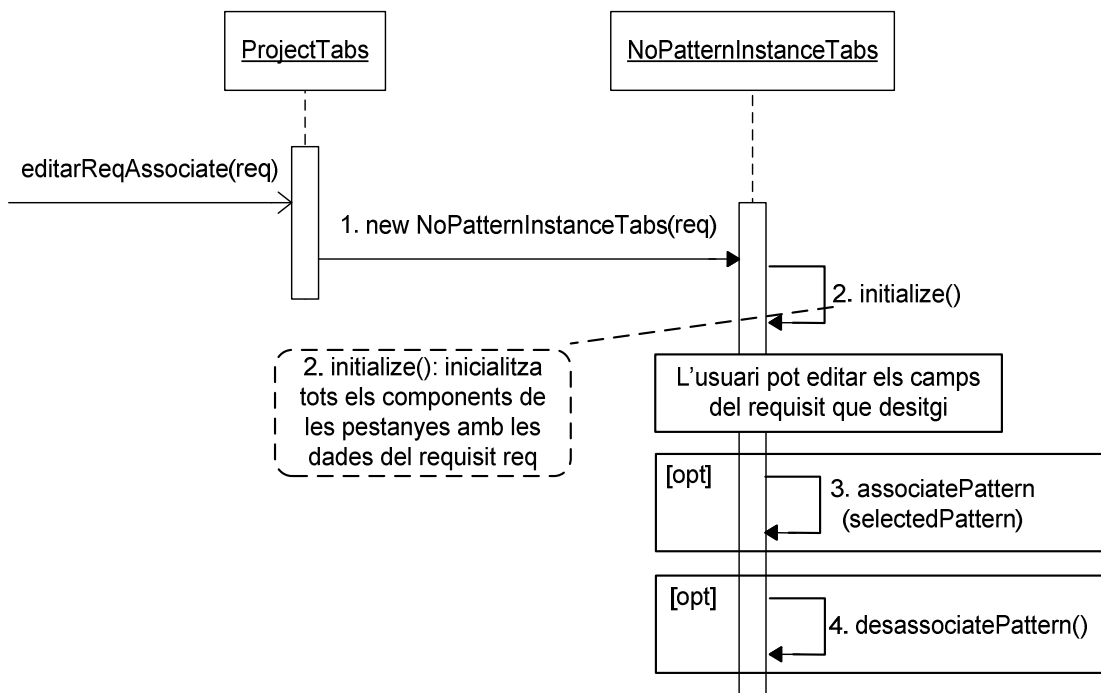


Figura 53: Diagrama de disseny 5 de 7 de "Crear / Editar un requisit amb patró associat"

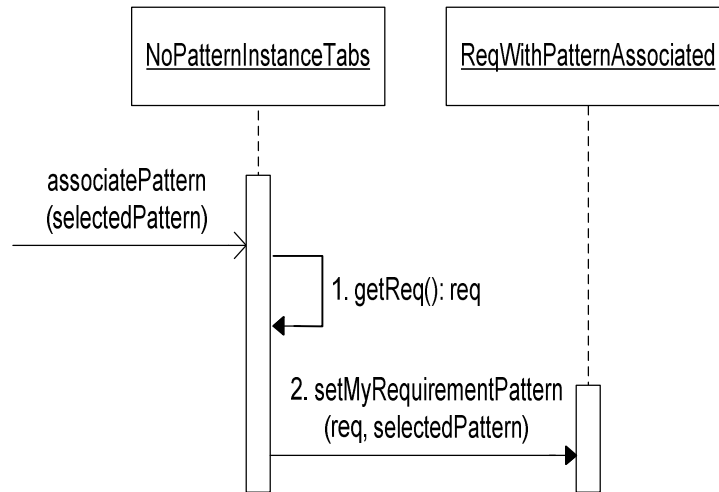


Figura 54: Diagrama de disseny 6 de 7 de "Crear / Editar un requisit amb patró associat"

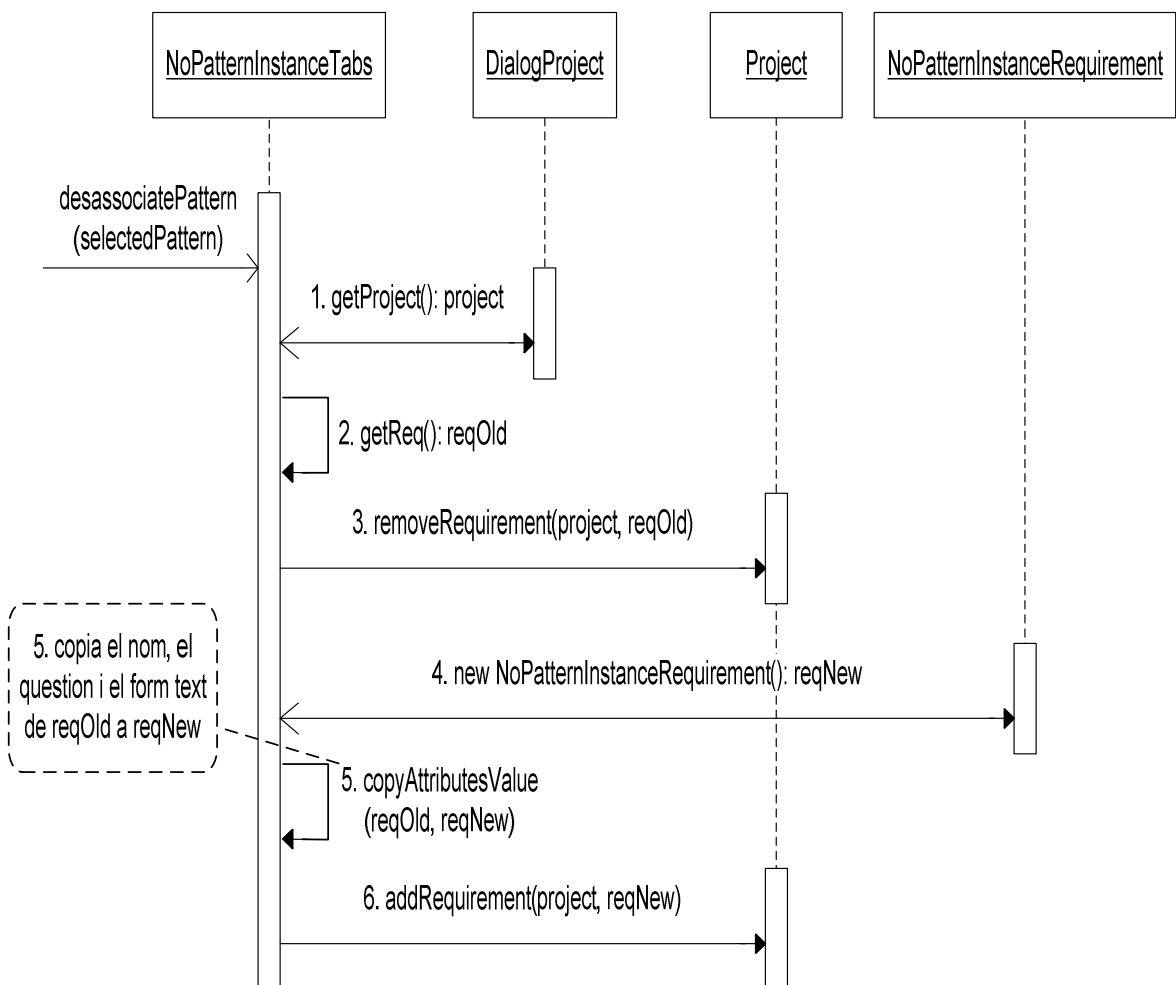


Figura 55: Diagrama de disseny 7 de 7 de "Crear / Editar un requisit amb patró associat"

18.2.4. Crear / Editar un requisit instància d'un patró

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten crear un nou requisit de tipus "instància d'un patró" dins d'un projecte o bé editar-lo.

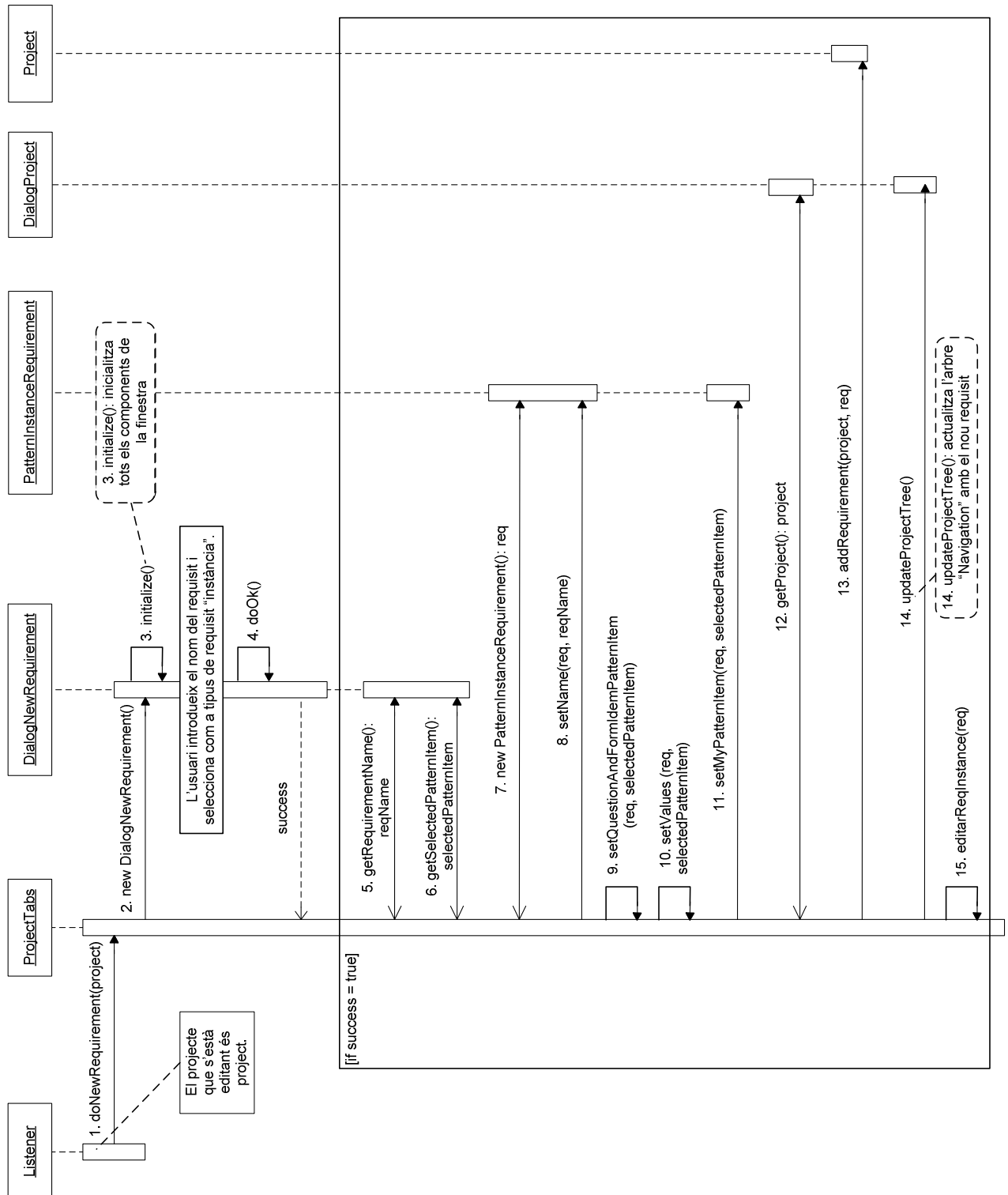


Figura 56: Diagrama de disseny 1 de 10 de "Crear / Editar un requisit instància d'un patró"

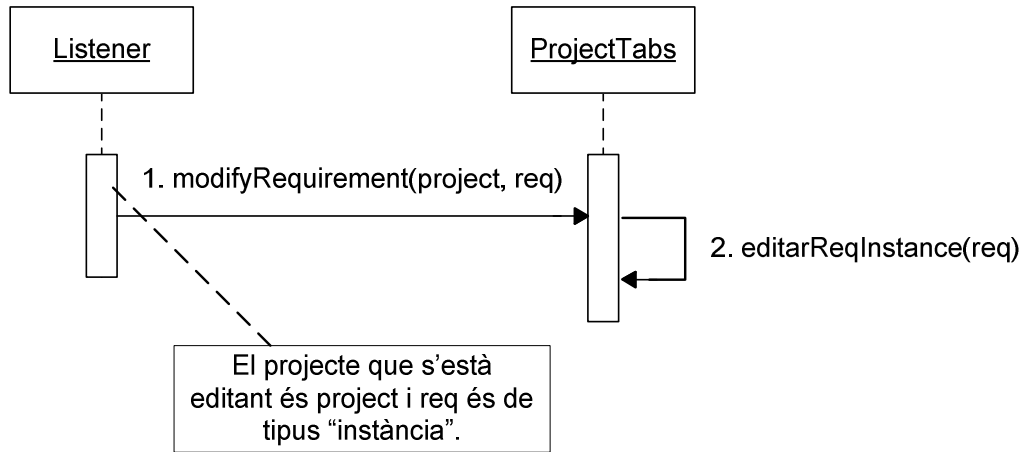


Figura 57: Diagrama de disseny 2 de 10 de "Crear / Editar un requisit instància d'un patró"

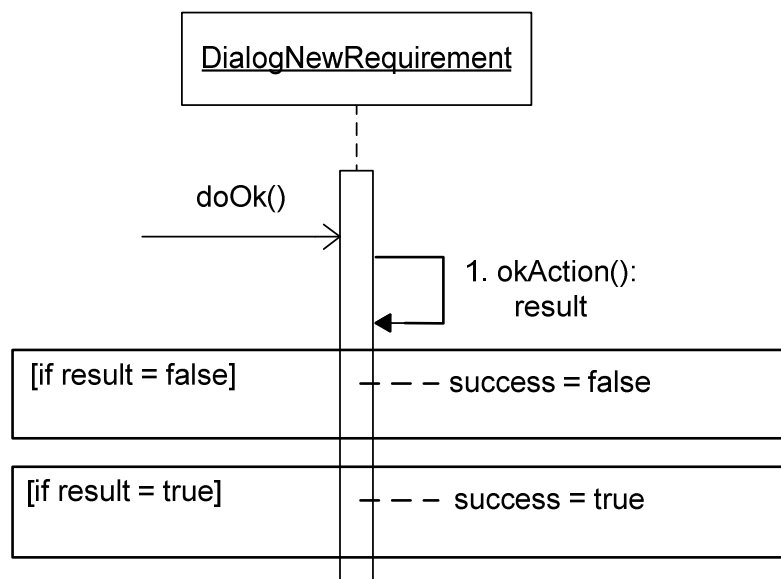


Figura 58: Diagrama de disseny 3 de 10 de "Crear / Editar un requisit instància d'un patró"

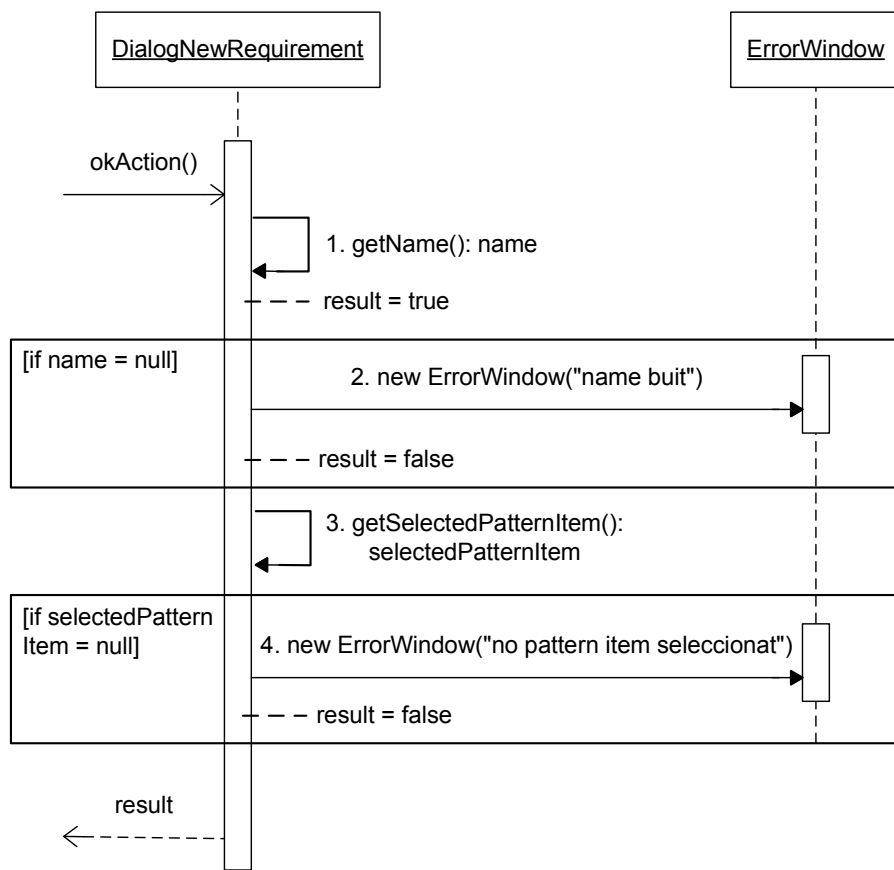


Figura 59: Diagrama de disseny 4 de 10 de "Crear / Editar un requisit instància d'un patró"

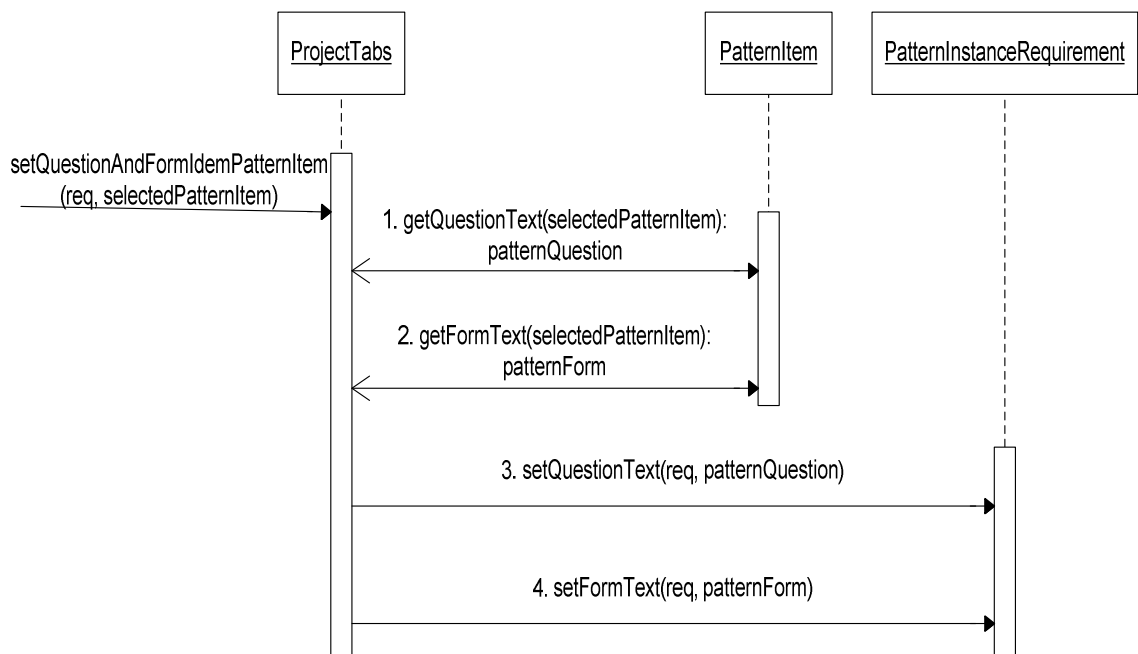


Figura 60: Diagrama de disseny 5 de 10 de "Crear / Editar un requisit instància d'un patró"

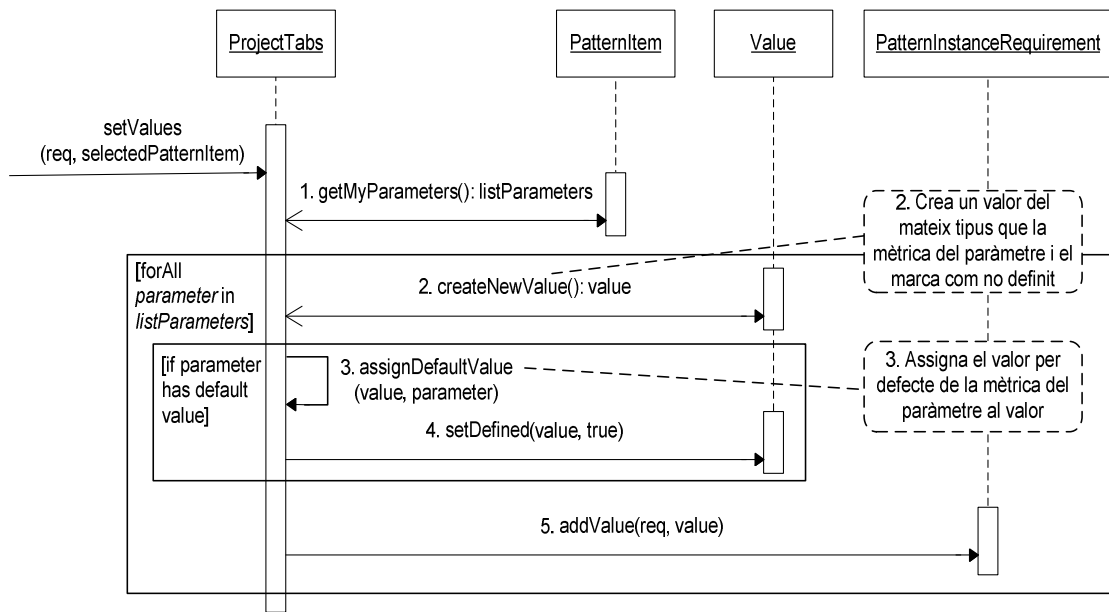


Figura 61: Diagrama de disseny 6 de 10 de "Crear / Editar un requisit instància d'un patró"

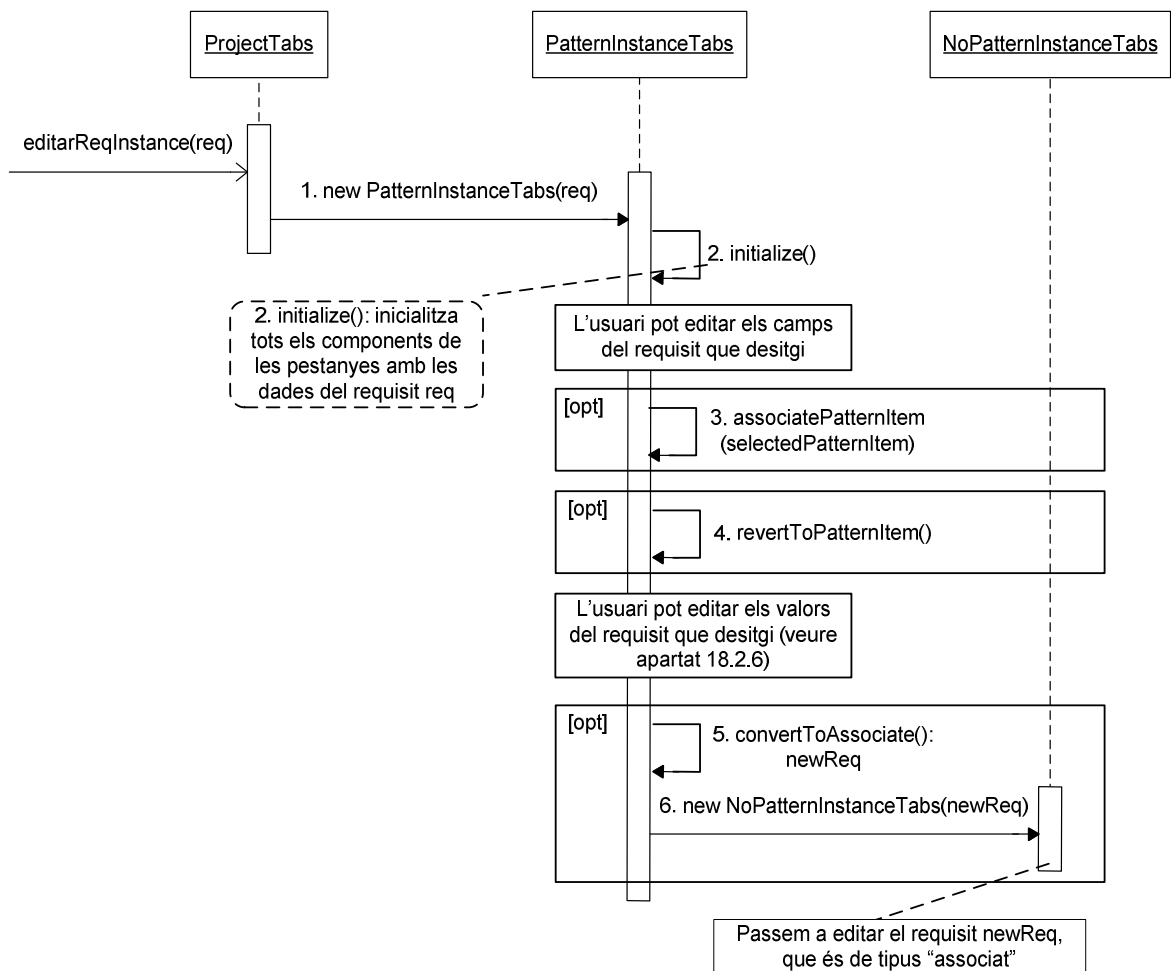


Figura 62: Diagrama de disseny 7 de 10 de "Crear / Editar un requisit instància d'un patró"

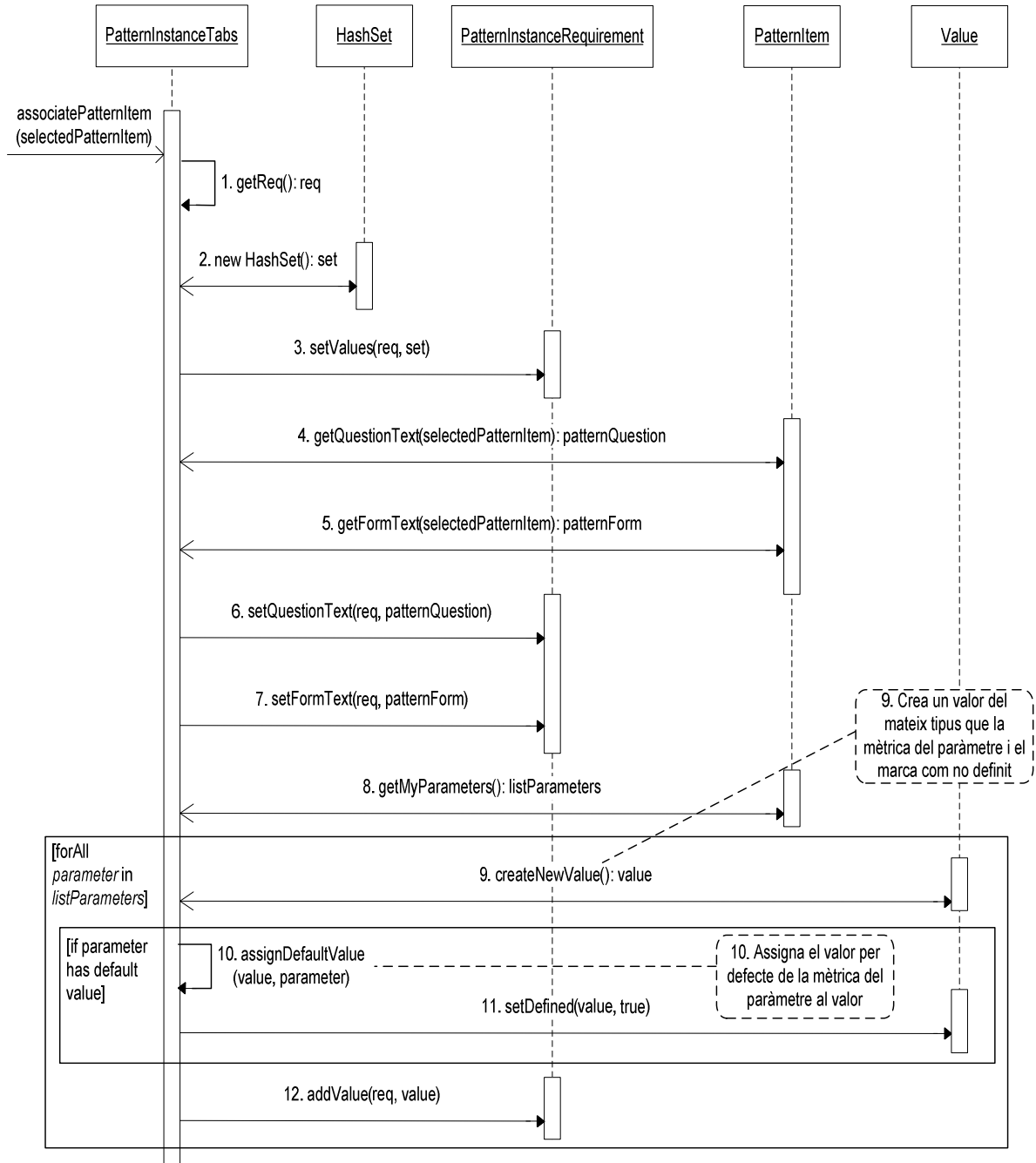


Figura 63: Diagrama de disseny 8 de 10 de "Crear / Editar un requisit instància d'un patró"

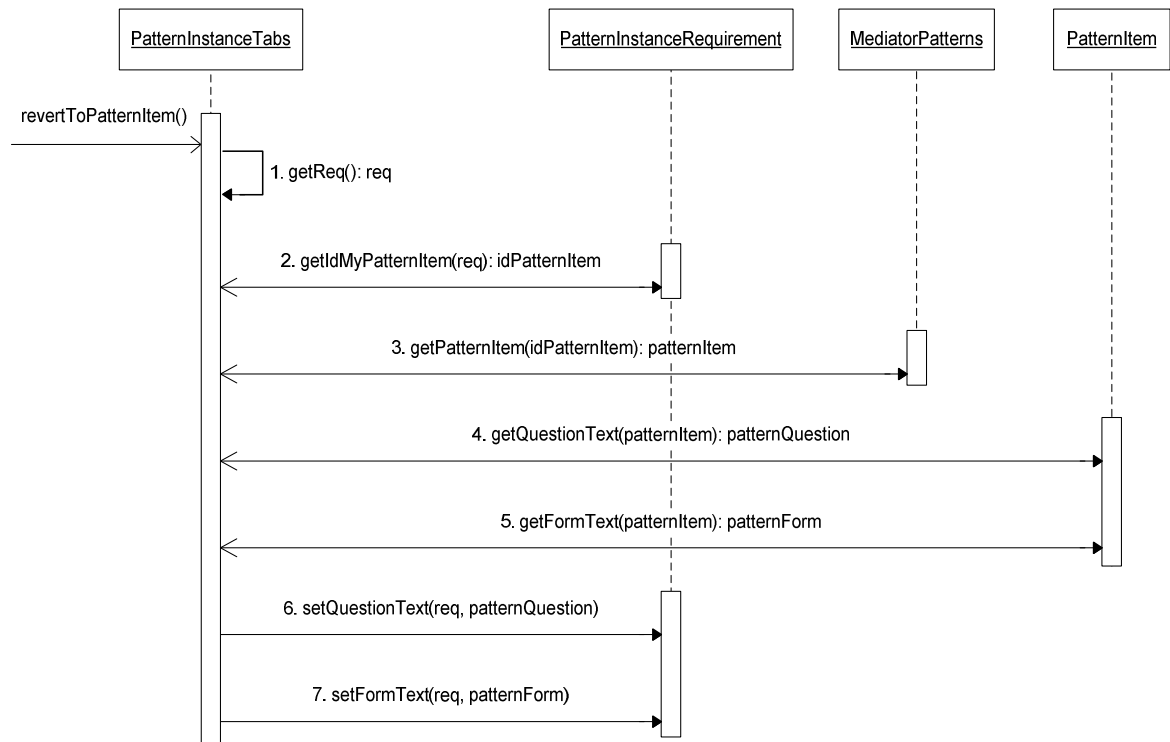


Figura 64: Diagrama de disseny 9 de 10 de "Crear / Editar un requisit instància d'un patró"

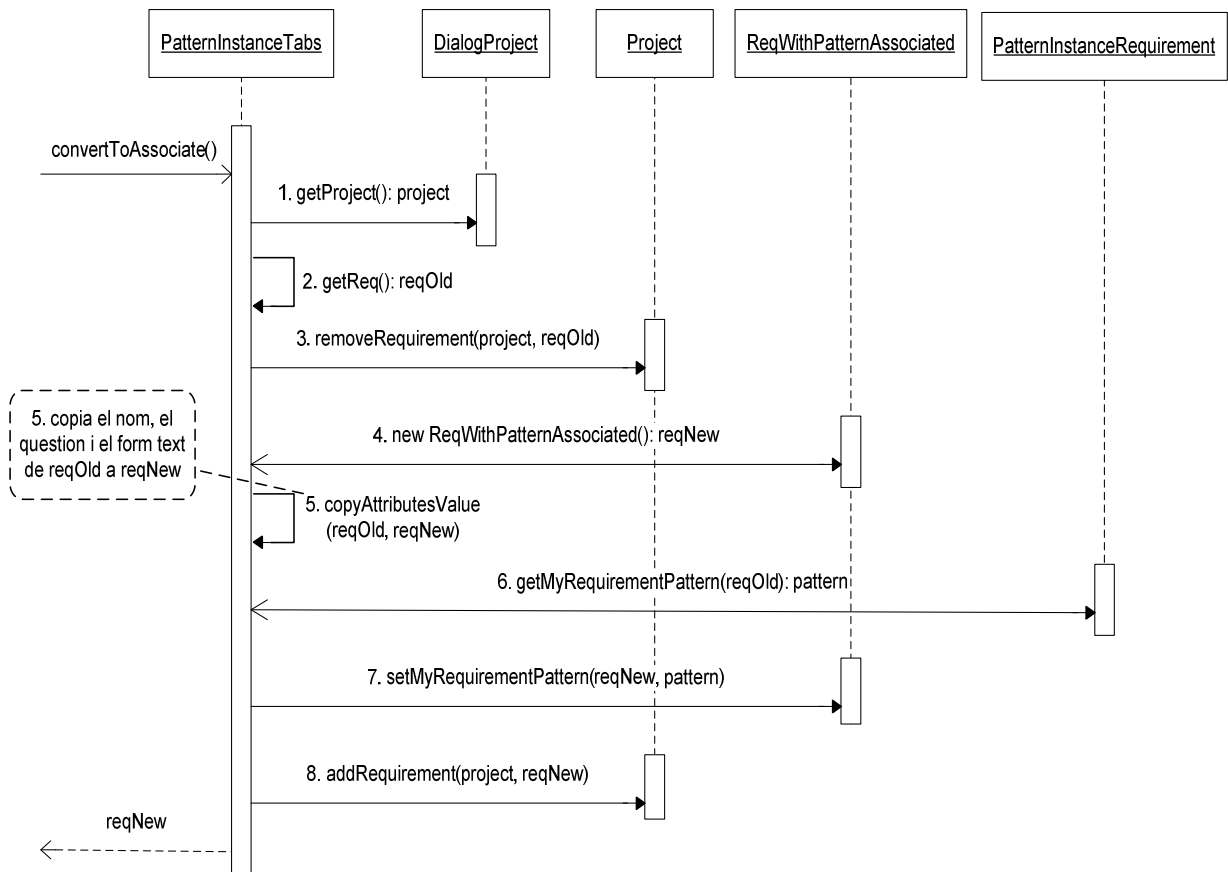


Figura 65: Diagrama de disseny 10 de 10 de "Crear / Editar un requisit instància d'un patró"

18.2.5. Eliminar un requisit d'un projecte

En el següent diagrama tenim la interacció entre classes amb les operacions necessàries que permeten eliminar un requisit d'un projecte. Cal tenir en compte que només esborrem un requisit quan hi ha un requisit seleccionat a la taula de requisits (és a dir, quan índex ≥ 0); en altre cas no fem res.

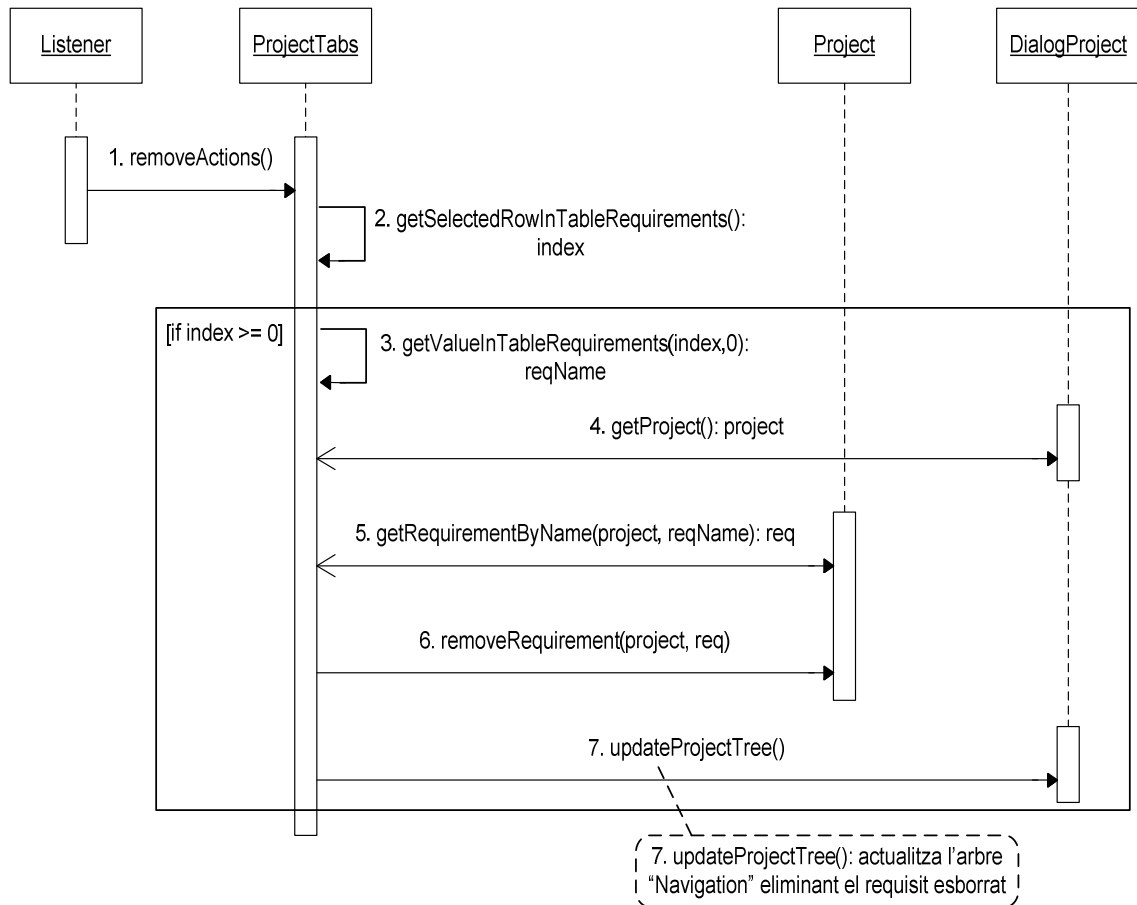


Figura 66: Diagrama de disseny de "Eliminar un requisit d'un projecte"

18.2.6. Modificar un valor d'un requisit

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten modificar un valor seleccionat d'un requisit que instància un patró. Per tal de no allargar massa el capítol de disseny, només representarem els diagrames de seqüència que corresponen al digrama de seqüència general quan es realitza l'acció de modificar un valor i el diagrama de seqüència de modificar un valor tipus integer.

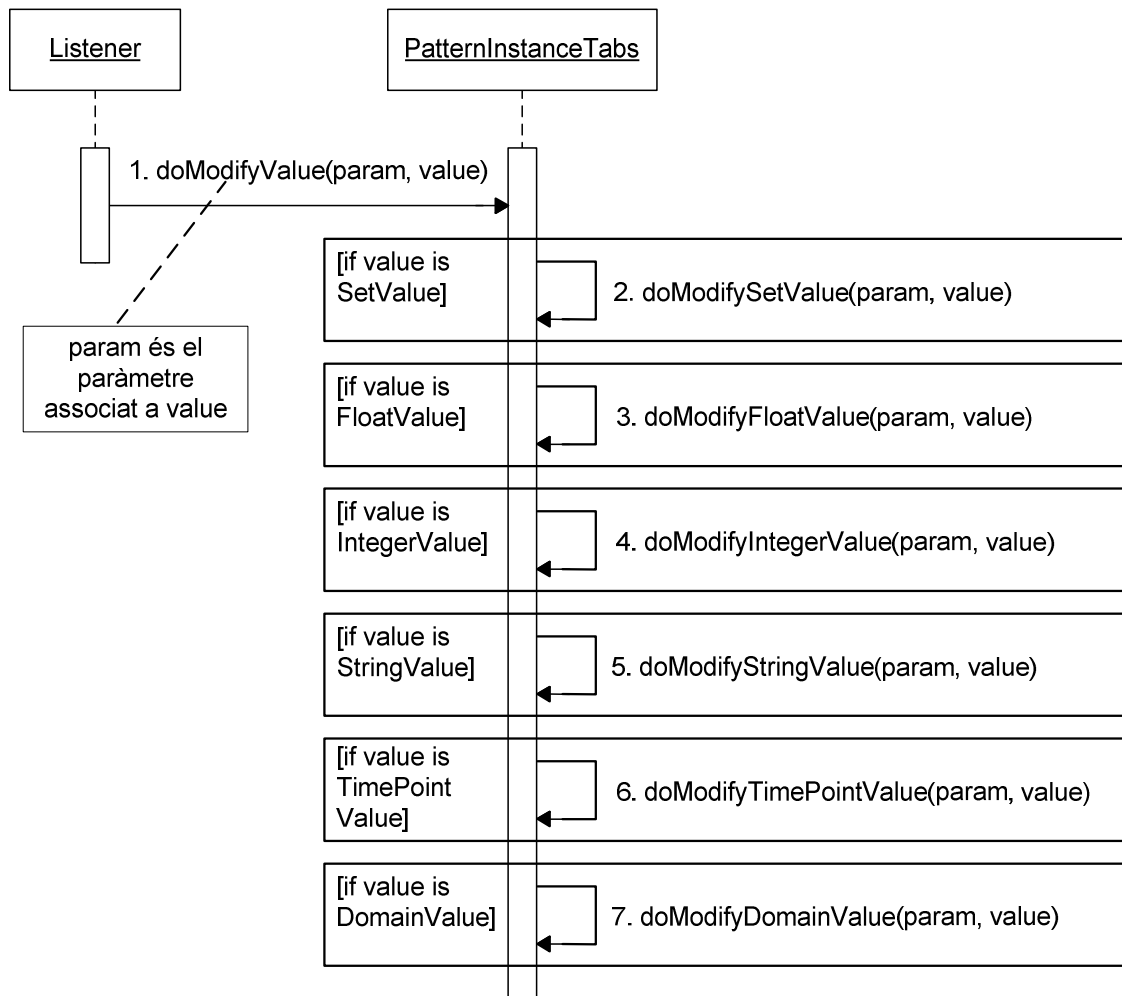


Figura 67: Diagrama de disseny 1 de 2 de "Modificar un valor d'un requisit"

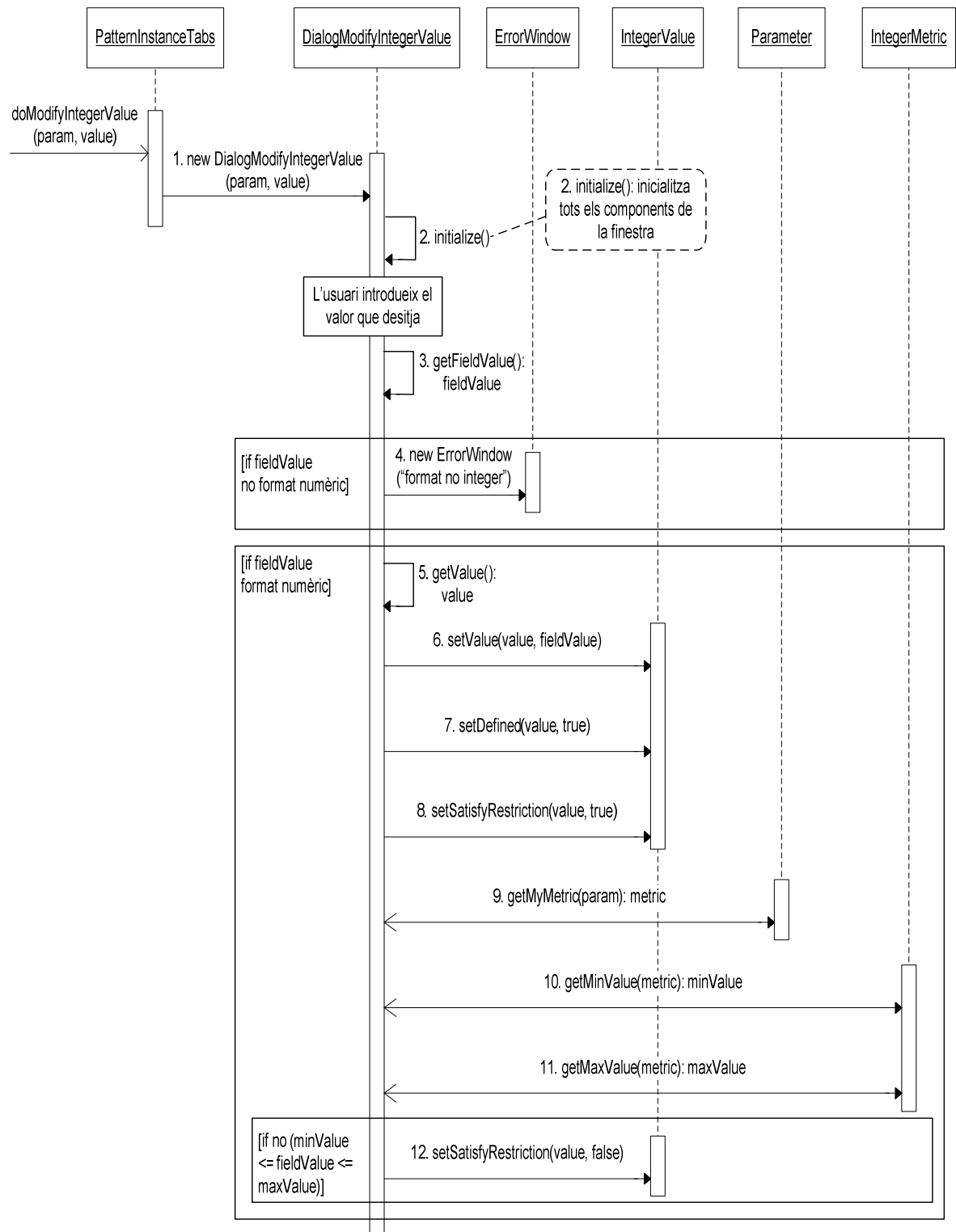


Figura 68: Diagrama de disseny 2 de 2 de "Modificar un valor d'un requisit"

18.3. Gestió de projectes

18.3.1. Copiar un projecte

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten copiar un projecte.

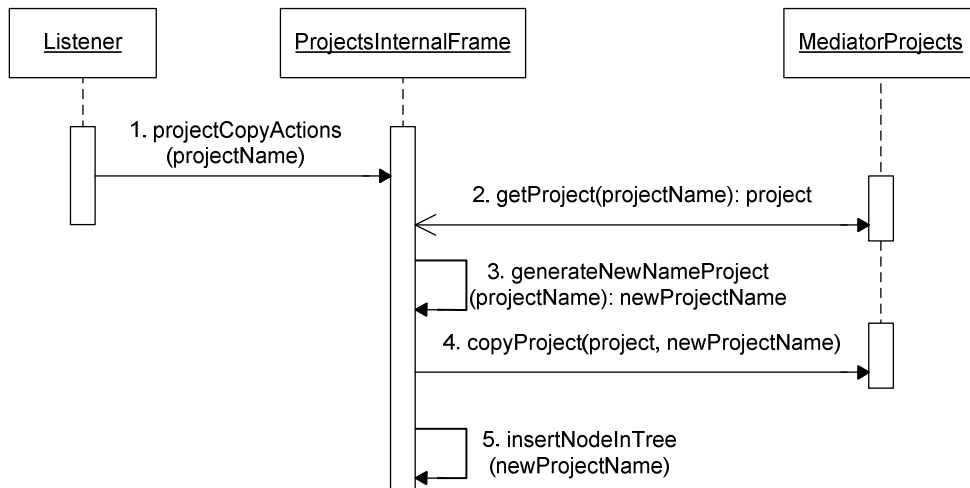


Figura 69: Diagrama de disseny 1 de 3 de "Copiar un projecte"

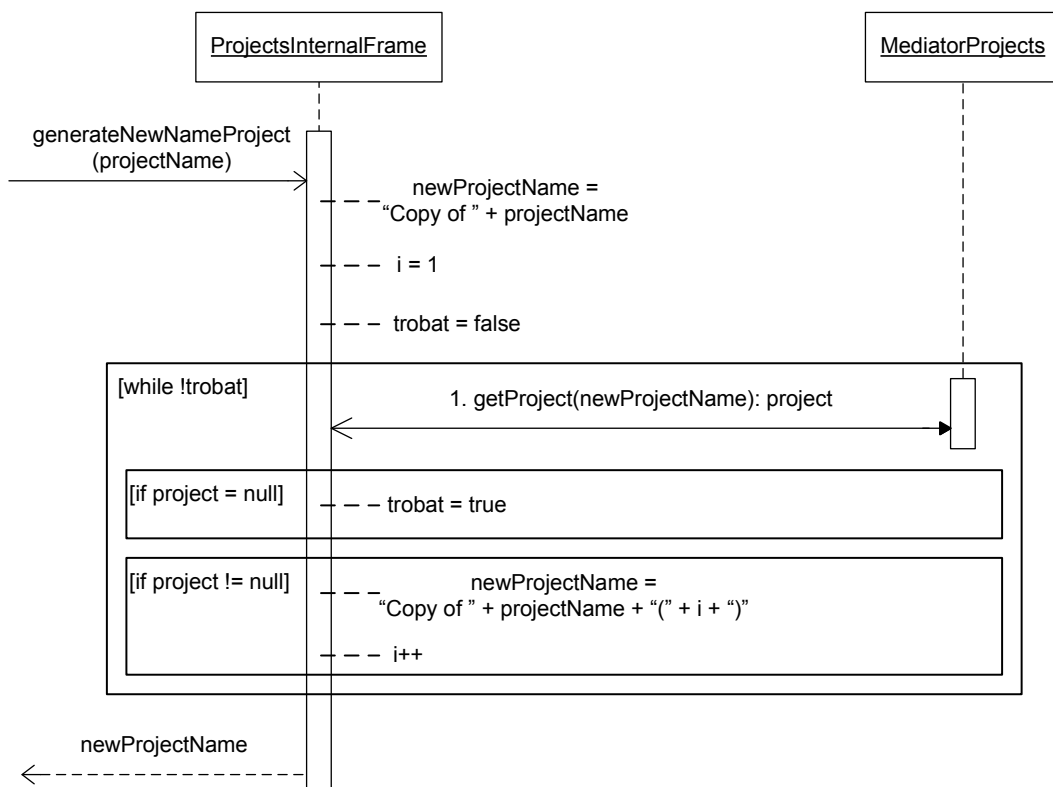


Figura 70: Diagrama de disseny 2 de 3 de "Copiar un projecte"

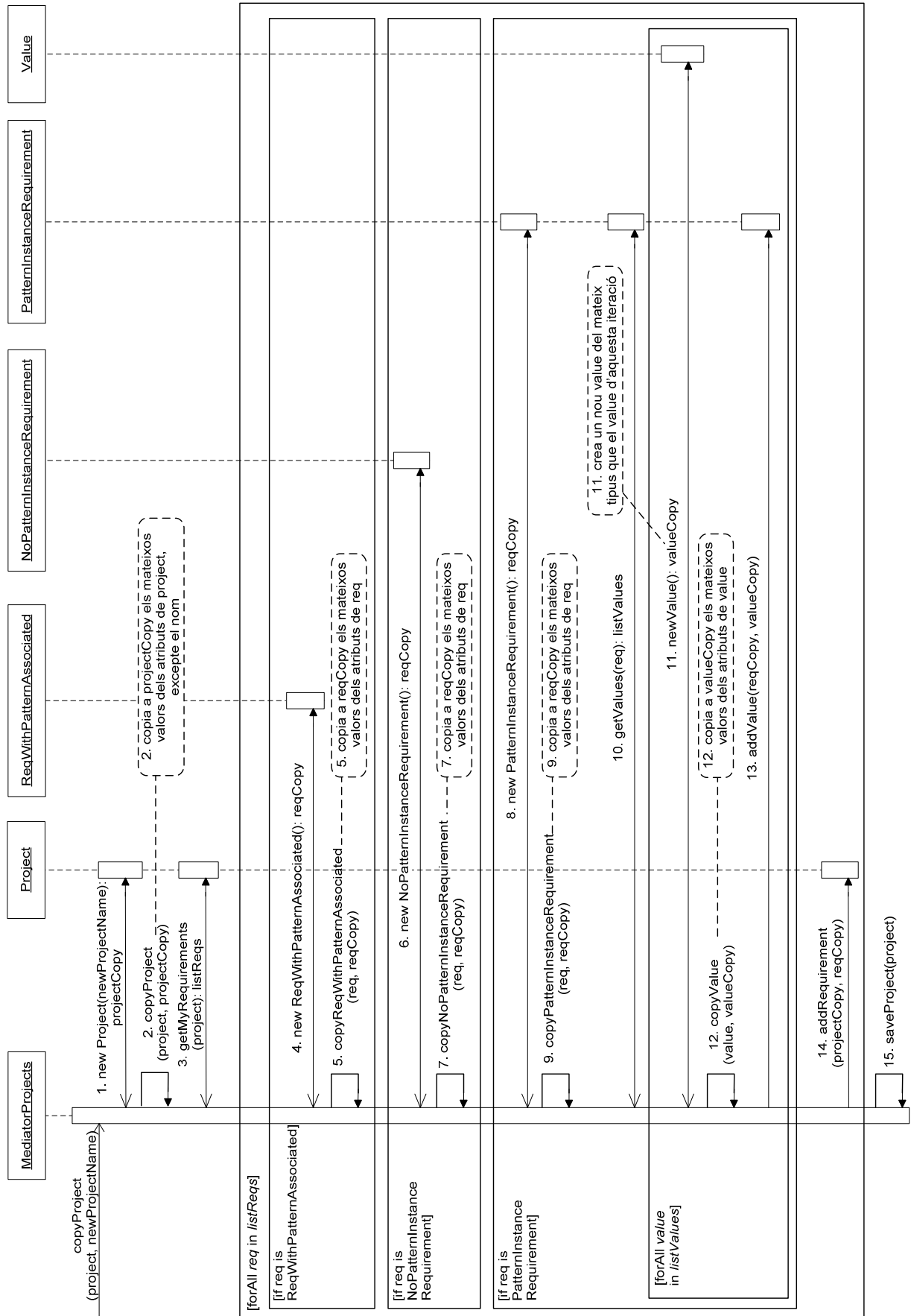


Figura 71: Diagrama de disseny 3 de 3 de "Copiar un projecte"

18.3.2. Eliminar un projecte

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten eliminar un projecte.

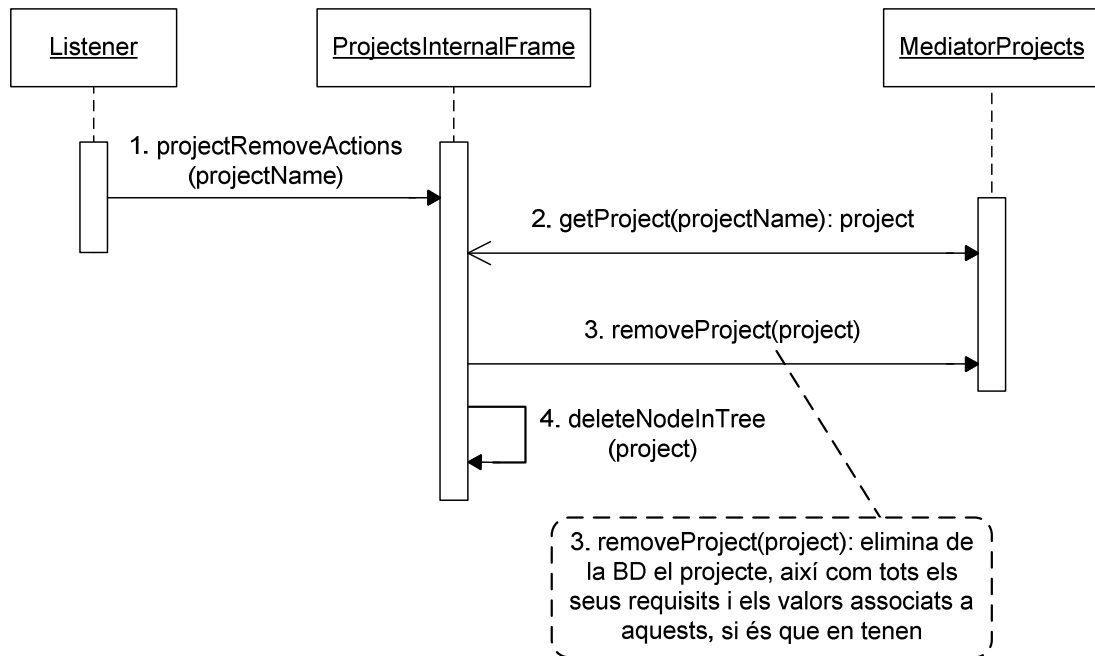


Figura 72: Diagrama de disseny de "Eliminar un projecte"

18.4. Interoperabilitat

18.4.1. Generar pdf amb tota la informació d'un projecte

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten generar un document pdf que conté tota la informació d'un projecte donat.

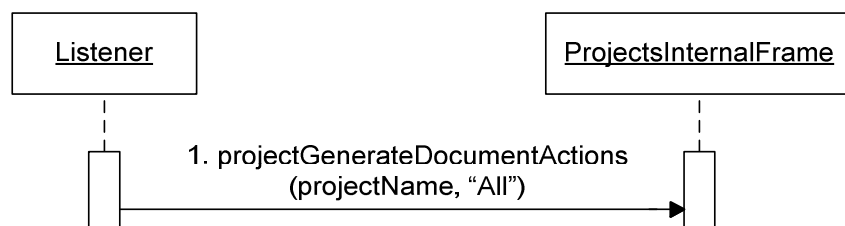


Figura 73: Diagrama de disseny 1 de 5 de "Generar pdf amb tota la informació d'un projecte"

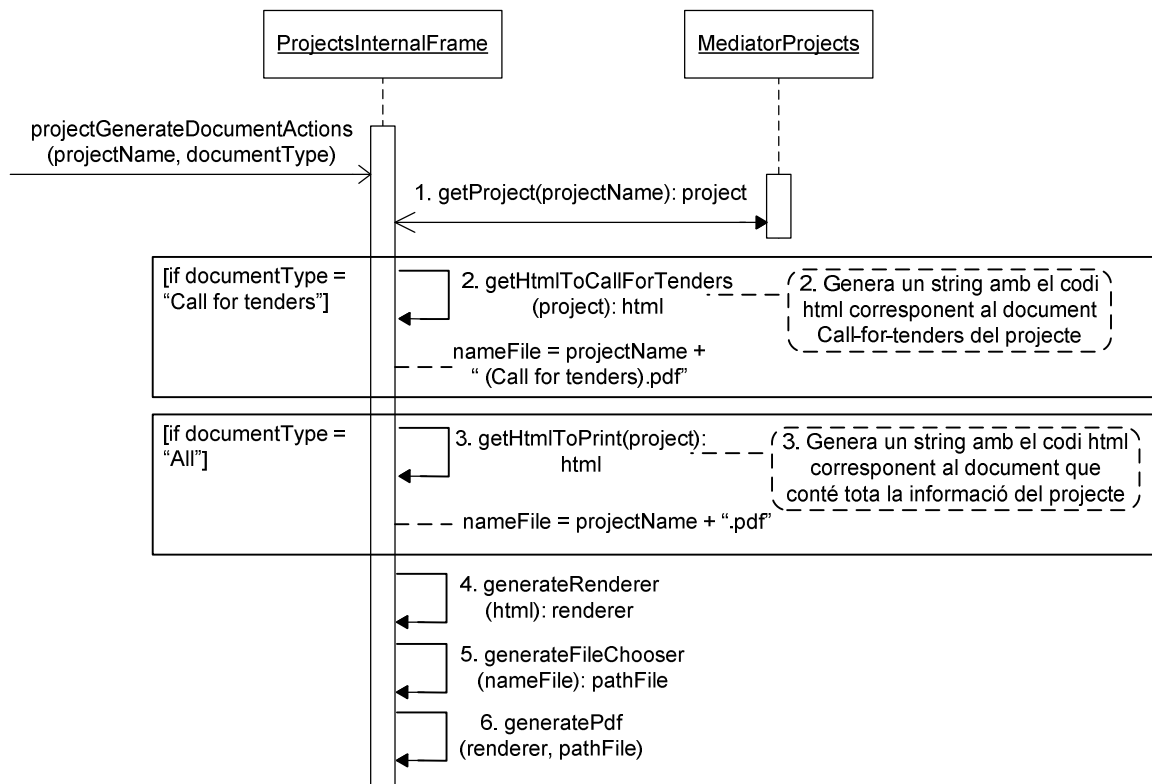


Figura 74: Diagrama de disseny 2 de 5 de "Generar pdf amb tota la informació d'un projecte"

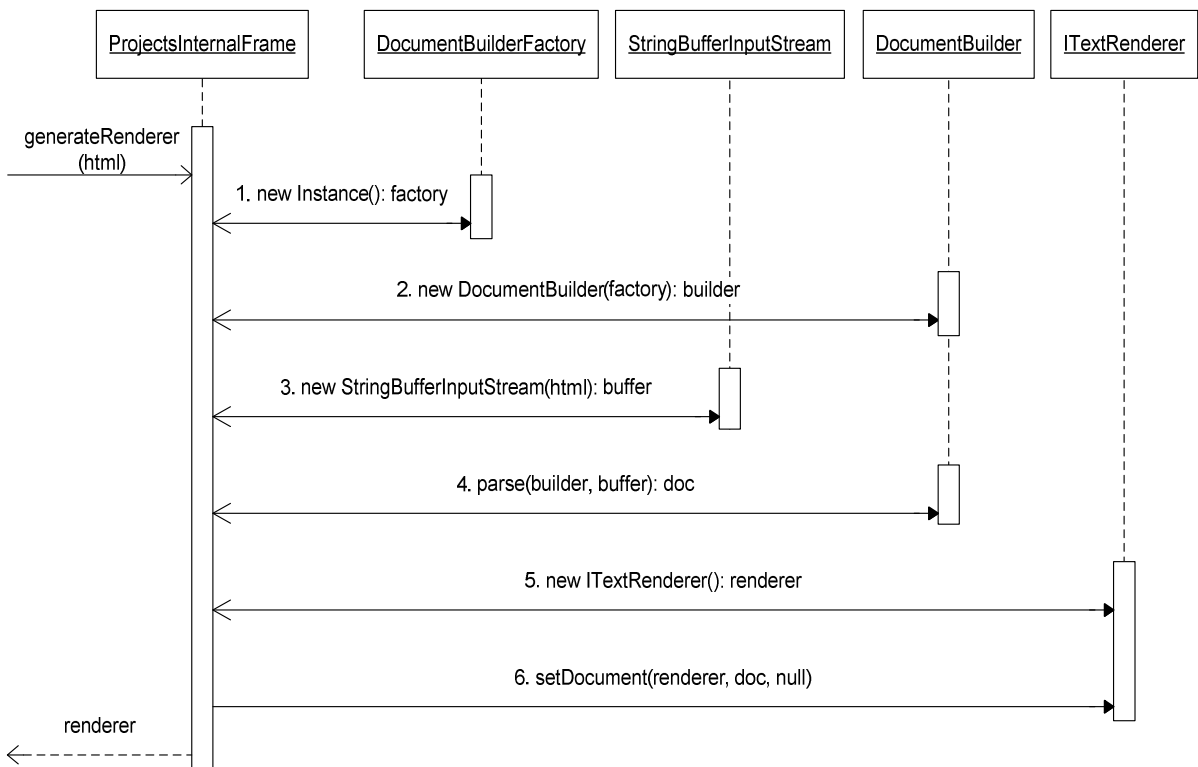


Figura 75: Diagrama de disseny 3 de 5 de "Generar pdf amb tota la informació d'un projecte"

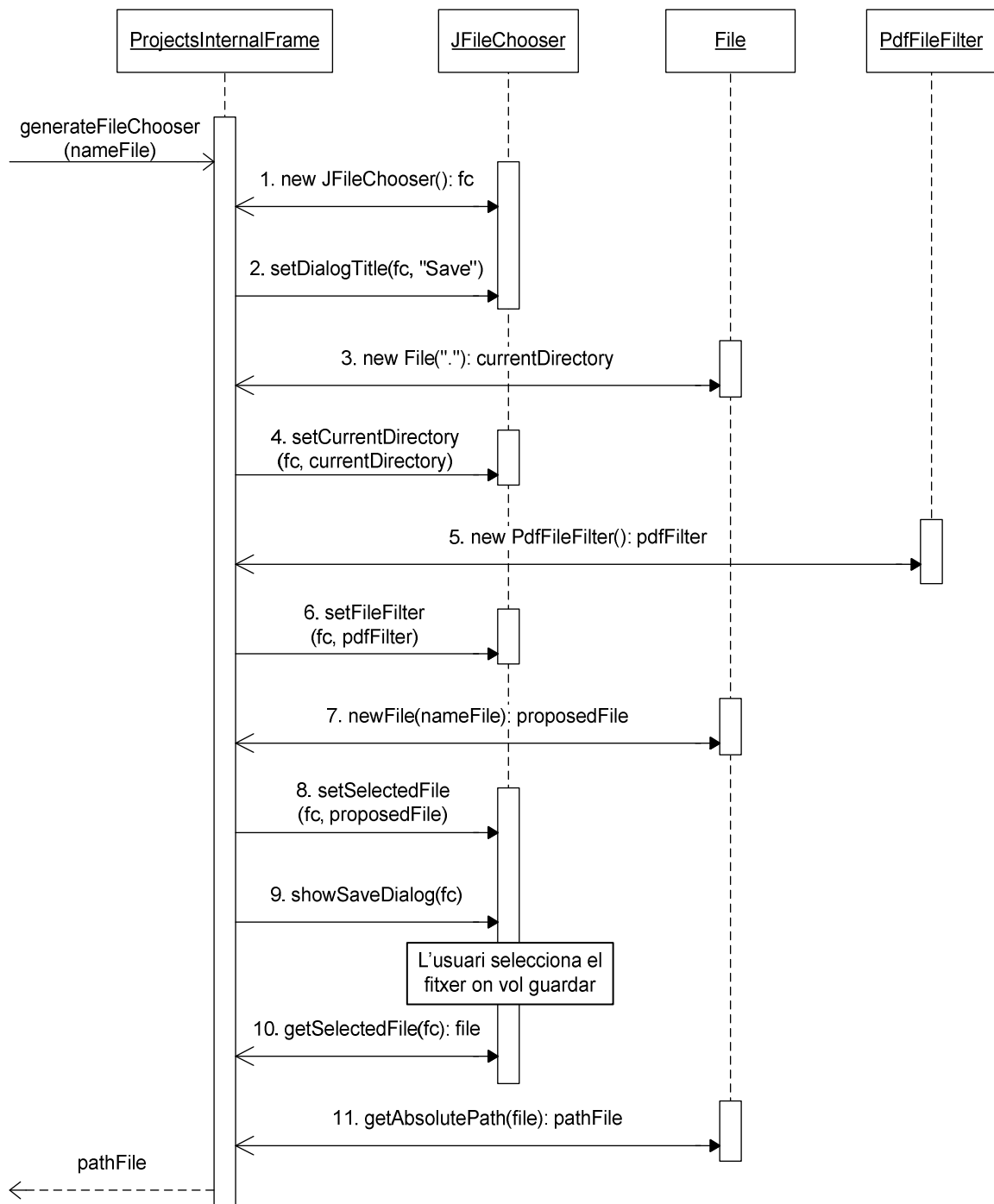


Figura 76: Diagrama de disseny 4 de 5 de "Generar pdf amb tota la informació d'un projecte"

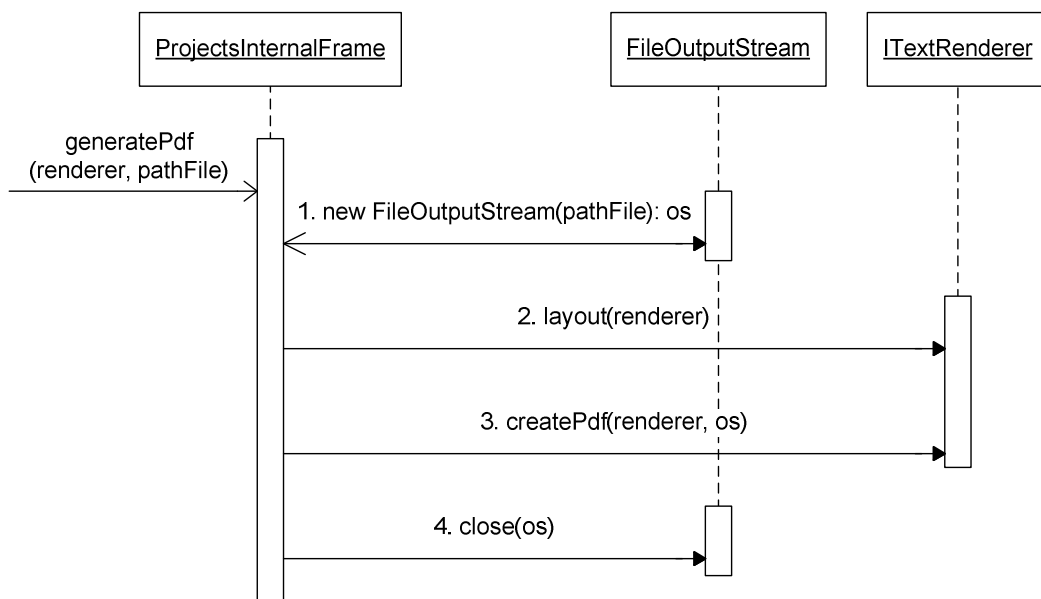


Figura 77: Diagrama de disseny 5 de 5 de "Generar pdf amb tota la informació d'un projecte"

18.4.2. Generar pdf amb el document call-for-tenders d'un projecte

En els següents diagrames tenim la interacció entre classes amb les operacions necessàries que permeten generar un document pdf que conté la informació del document call-for-tenders d'un projecte donat.

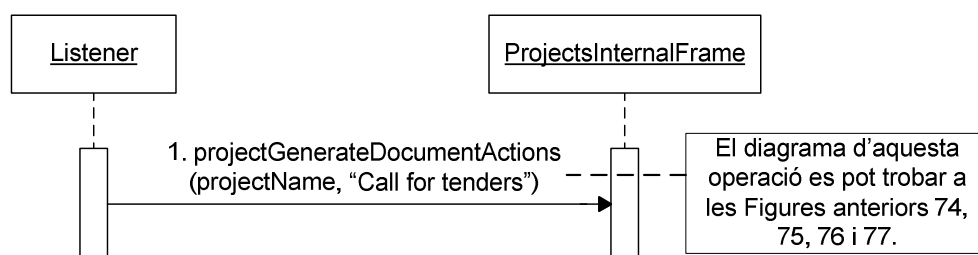


Figura 78: Diagrama de disseny de "Generar pdf amb el document call-for-tenders d'un projecte"

19. Proves del Software

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager.	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
19.1. Introducció	240
19.2. Comprovació de les funcionalitats en el curs normal d'esdeveniments.....	240
19.2.1. Taula de comprovacions	240
19.2.2. Llista de correccions.....	241
19.2.3. Taula final de comprovacions	241
19.3. Comprovació de les restriccions d'integritat.....	242
19.3.1. Taula de comprovacions	242
19.3.2. Llista de correccions.....	242
19.3.3. Taula final de comprovacions	243
 20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

19.1. Introducció

Aquest artefacte té com a objectiu la comprovació del bon funcionament de totes les operacions de l'eina. Serveix per verificar i validar totes les funcionalitats de l'aplicació de manera formal.

A tal efecte, s'ha comprovat la resposta de l'aplicació en els diferents casos d'ús tant per la visualització en la interfície (que es mostrés bé l'arbre i els camps dels diàlegs) com per la informació emmagatzemada a la Base de Dades. S'ha comprovat també el comportament del curs normal d'esdeveniments de tots els casos d'ús així com també que es garanteixin les restriccions d'integritat quan aquestes s'intentin violar, com queda especificat en els casos alternatius del mateix cas d'ús.

19.2. Comprovació de les funcionalitats en el curs normal d'esdeveniments

19.2.1. Taula de comprovacions

En aquesta taula es mostren els casos d'ús implementats (capítol 12) amb la resposta de l'aplicació tant per la visualització en la interfície (que es mostrés bé l'arbre i els camps dels diàlegs) com per la informació emmagatzemada a la Base de Dades. En ella es mostra una senyal en forma de *V* sí la comprovació s'ha de tenir en compte en el cas d'ús i funciona correctament, altrament es mostrarà un identificador Ci en el que s'explicarà en l'apartat següent l'error trobat i la solució aportada en aquesta fase de Proves del Software. Per últim tots els símbols “---” indiquen que en el cas d'ús no s'ha de tenir en compte la comprovació.

Cas d'ús	Visualització	BBDD	Cas d'ús	Visualització	BBDD
1.1	✓	---	2.3.3	✓	✓
1.2	✓	---	2.4.1	C ₂	✓
1.3	✓	---	2.4.2	C ₂	✓
1.4	✓	---	2.4.3	C ₂	✓
1.5	✓	---	2.5	✓	✓
1.6	✓	---	2.6	✓	✓
2.1	✓	✓	3.1	✓	✓
2.2	C ₁	✓	3.2	✓	C ₃
2.3.1	✓	✓	4.1	---	✓
2.3.2	✓	✓	4.2	---	✓

Taula 8: Resultats inicials de les proves del software, casos d'ús

19.2.2. Llista de correccions

C₁:

Quan s'editava el nom del projecte i es guardaven correctament els canvis a la Base de Dades, el nom del projecte no s'actualitzava correctament a l'arbre on es visualitzen els noms de tots els projectes. Només feia falta cridar a l'operació que modifica en un node de l'arbre el nom del projecte antic pel nou nom.

C₂:

Quan s'editava un requisit, fos del tipus que fos aquest, els seus atributs no s'actualitzaven correctament a la taula on es visualitzen tots els requisits del projecte. Feia falta crear un mètode que actualitzés a la taula esmentada el requisit editat i cridar-lo després d'editar el requisit.

C₃:

Al eliminar un projecte del repositori, quan algun dels seus requisits tenia un valor de tipus set, els valors simples que formaven part d'ell no s'eliminaven de la Base de Dades. Per tal de corregir-ho, s'ha modificat el mètode que eliminava un valor de tipus set de la Base de Dades per tal que també eliminés el seus valors simples.

19.2.3. Taula final de comprovacions

Cas d'ús	Visualització	BBDD	Cas d'ús	Visualització	BBDD
1.1	✓	---	2.3.3	✓	✓
1.2	✓	---	2.4.1	✓	✓
1.3	✓	---	2.4.2	✓	✓
1.4	✓	---	2.4.3	✓	✓
1.5	✓	---	2.5	✓	✓
1.6	✓	---	2.6	✓	✓
2.1	✓	✓	3.1	✓	✓
2.2	✓	✓	3.2	✓	✓
2.3.1	✓	✓	4.1	---	✓
2.3.2	✓	✓	4.2	---	✓

Taula 9: Resultats finals de les proves del software, casos d'ús

19.3. Comprovació de les restriccions d'integritat

19.3.1. Taula de comprovacions

En aquesta taula es mostren els casos d'ús implementats (capítol 12) amb les restriccions d'integritat de l'eina (capítol 13.2). En ella es mostra una senyal en forma de *V* sí la comprovació de la restricció d'integritat que s'ha de tenir en compte en el cas d'ús funciona correctament, altrament es mostrarà un identificador *C_i* en el que s'explicarà en l'apartat següent l'error trobat i la solució aportada en aquesta fase de Proves del Software. Per últim tots els símbols “-” indiquen que el cas d'ús no afecta ni es afectat per res per la restricció d'integritat en qüestió.

Cas d'ús	Restricció d'Integritat (RI)																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.1	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
2.2	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
2.3.1	-	-	✓	C ₁	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.3.2	-	-	✓	C ₁	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.3.3	-	-	✓	C ₁	-	✓	✓	✓	-	-	-	-	-	-	-	✓	-	✓	C ₂
2.4.1	-	-	✓	C ₁	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.4.2	-	-	✓	C ₁	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.4.3	-	-	✓	C ₁	-	✓	C ₃	C ₃	-	-	-	-	-	-	-	✓	-	✓	C ₂
2.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.6	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	-	C ₄	-	-
3.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Taula 10: Resultats inicials de les proves del software, restriccions d'integritat

19.3.2. Llista de correccions

C₁:

Quan es creava/editava un requisit es permetia afegir el nom d'un requisit que ja existís dins del projecte. Per corregir-ho, es va crear un mètode que comprovava abans d'afegir/modificar el requisit dins del projecte si ja existia cap altre requisit amb el mateix nom en el projecte en qüestió.

C₂:

Quan l'atribut formText d'un requisit que instància un patró no contenia el nom de tots els paràmetres del Pattern Item que instanciava, el sistema no responia amb error però tampoc deixava guardar els canvis. El problema venia provocat per que l'error que es llençava quan això passava no era recollit mai per cap mètode. El problema s'ha resolt satisfactòriament sense dificultats.

C₃:

Al editar un requisit que instància un patró, el Requirement Pattern i Requirement Form associats no s'actualitzaven correctament i això provocava que l'aplicació fallés quan, un cop modificat el Pattern Item que instanciava el requisit, s'intentava tornar a editar el mateix requisit. Només feia falta actualitzar correctament el Requirement Pattern i el Requirement Form associats al requisit quan es modificava el seu Pattern Item per solucionar els errors.

C₄:

Al modificar un valor de tipus string es deixava que aquest fos un valor buit o un conjunt d'espais en blanc perquè faltava la comprovació que impedís que això passés.

19.3.3. Taula final de comprovacions

Cas d'ús	Restricció d'Integritat (RI)																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.1	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
2.2	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-
2.3.1	-	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.3.2	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.3.3	-	-	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-	✓	-	✓	✓
2.4.1	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.4.2	-	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	✓	-	✓	-
2.4.3	-	-	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-	✓	-	✓	✓
2.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2.6	-	-	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-
3.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Taula 11: Resultats finals de les proves del software, restriccions d'integritat

20. Planificació i Execució del Projecte

1.	Introducció	11
2.	Patrons de requisits	15
3.	Eines de Treball	29
4.	Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager		39
5.	Manteniment i Extensió de Software	41
6.	Anàlisi de la 1 ^a versió de RPToolManager	47
7.	Model Conceptual	61
8.	Base de Dades	67
9.	Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager		79
10.	Metodologia: Àgil	81
11.	Anàlisi de Requeriments	85
12.	Model de Casos d'ús	97
13.	Model Conceptual	125
14.	Model de Comportament	133
15.	Disseny de la Interfície	167
16.	Reutilització de Components	189
17.	Base de Dades	195
18.	Document de Disseny	201
19.	Proves del Software	239
20.	Planificació i Execució del Projecte	245
20.1.	Planificació	246
20.2.	Execució	248
21.	Future Work	251
22.	Conclusions	255
23.	Bibliografia	257
24.	Contingut del CD	261
ANNEX 1: Modificacions al codi font de RPToolManager		265

20.1. Planificació

En aquest capítol parlarem de la planificació seguida durant tot el projecte, citant les diferents etapes que aquest ha tingut.

El tema del projecte es va decidir a finals de Juny de 2009, moment en el que es va inscriure, i al Juliol del mateix any es va matricular amb la intenció de llegir el projecte a Gener de 2010. Tot i que no sol ser habitual realitzar el projecte en només un quadrimestre, en el meu cas no era cap problema ja que podia dedicar tot el dia al mateix. Pel que fa a “l’horari de treball”, vaig decidir dedicar una mitja de 8 hores diàries de dilluns a divendres i els cap de setmana una mitja de 5 hores per dia, sense tenir en compte en ambdós casos els dies festius. Així doncs, setmanalment dedicava al projecte una mitja de 50 hores. Aquest horari és el que es va realitzar des de mitjans de Setembre fins al final del projecte.

Per tal de planificar correctament el projecte, el primer que vaig fer va ser descomposar tota la feina que tenia per fer en tasques, de manera que a cada una d’elles li pogués assignar un període determinat de temps. Així doncs, el projecte es va dividir en 6 tasques diferents, que són:

1. Estat de l’art sobre Patrons de Requisits

Tot i que disposava d’una petita base de coneixement sobre els Patrons de Requisits, era necessari aprofundir més en el tema. Per això, durant l’estiu de 2009 (Juliol i Agost) vaig llegir diversos articles i treballs relacionats ([RMFQ09], [S05], [MRRH05], [CJ07], [LMV97], [CR00], [R96] i [DBRT99]) i alguns capítols del llibre [W07].

2. Instal·lació de l’entorn de treball

Aquesta fase del projecte va consistir en instal·lar l’Eclipse, configurar-lo i fer que el codi font de l’eina *RPToolManager* compilés i s’executés sense problemes en l’entorn de desenvolupament.

3. Anàlisi de l’aplicació RPToolManager

Donat que no coneixia l’eina *RPToolManager*, la primera part d’aquesta tasca va consistir en estudiar l’eina i parlar amb un dels desenvolupadors d’aquesta, l’Oriol Collell, per tal de tenir una idea global de com funcionava l’eina i saber de quina manera estava organitzat el codi d’aquesta. A continuació, partint d’un anàlisi de l’aplicació realitzat per l’Oriol Collell, es va fer un altre anàlisi de major profunditat de l’eina, detectant així la gran part d’errors que aquesta tenia. Una petita part de les mancances i de les noves funcionalitats a afegir a *RPToolManager* es van aconseguir també durant aquesta fase d’anàlisi; la resta de mancances i noves funcionalitats es van obtenir mitjançant diverses reunions amb la client de l’aplicació, la Carme Quer.

4. Desenvolupament de la 2^a versió de RPToolManager

Un cop realitzat l'anàlisi de l'eina, es va desenvolupar la 2^a versió d'aquesta. El procediment a seguir va ser, per cada error, mancança o nova funcionalitat detectat durant l'anàlisi, implementar-lo i a continuació testear-lo. Un cop ja es va tenir la 2^a versió de l'eina, es van tornar a testear totes les modificacions i noves funcionalitats realitzades per tal d'assegurar el seu bon funcionament.

5. Desenvolupament de la 1^a versió de RPProjectManager

Seguint els principis de la metodologia àgil (vegis capítol 10), pel desenvolupament de la 1^a versió de *RPProjectManager* es va utilitzar un desenvolupament en espiral amb cicles molt curts. Cada grup de funcionalitats correspon a un cicle, de manera que es van obtenir 4 cicles. Les etapes de cada cicle són les clàssiques: Especificació, Disseny, Codificació i Testeig.

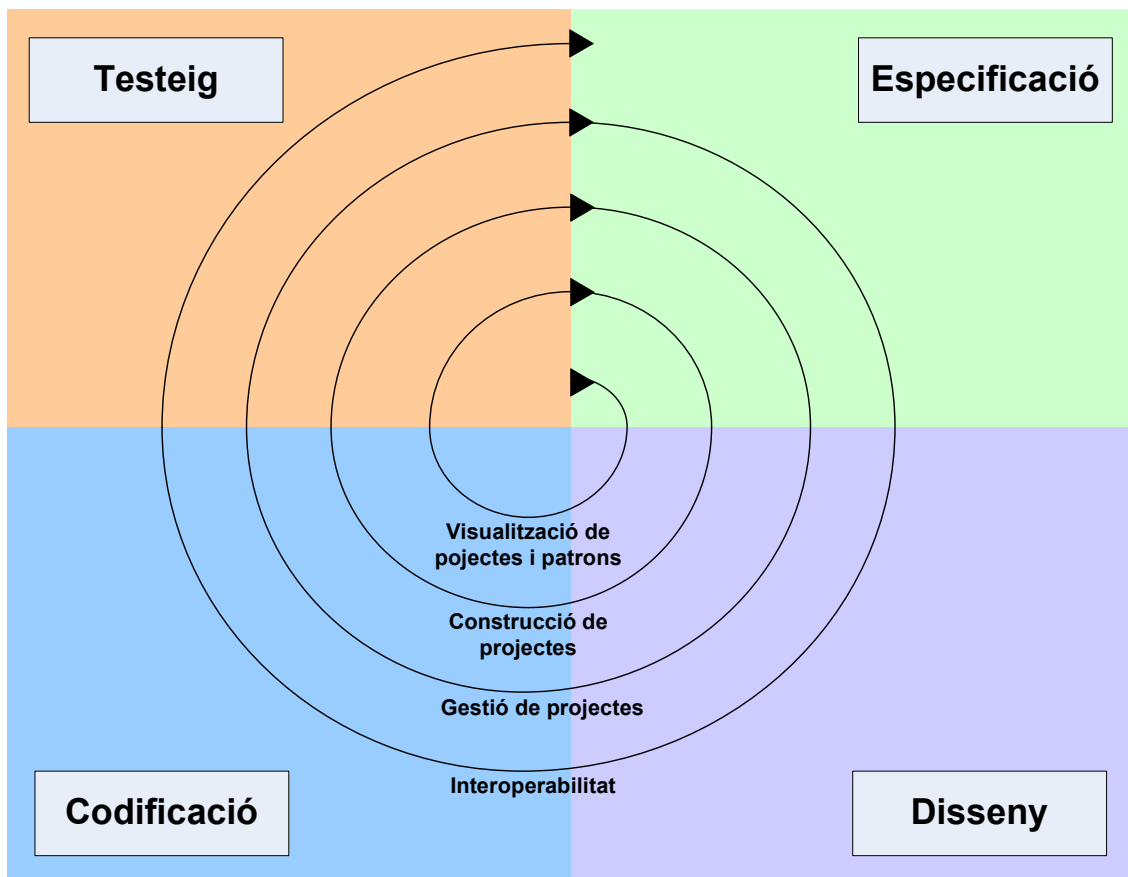


Figura 79: Cicles realitzats pel desenvolupament de la 1^a versió de RPProjectManager

6. Elaboració de la memòria

Per acabar el projecte cal realitzar la memòria d'aquest. Tot i que hi ha parts que es van realitzar durant les fases anteriors (realització dels models conceptuals, casos d'ús, etc.), no és fins aquesta etapa que es comença a redactar i construir el document en si mateix.

El temps estimat per cada activitat va ser el que segueix:





Activitat	Temps estimat
1. Estat de l'art sobre Patrons de Requisits	100h
2. Instal·lació de l'entorn de treball	100h
3. Anàlisi de l'aplicació <i>RPToolManager</i>	
4. Desenvolupament de la 2 ^a versió de <i>RPToolManager</i>	350h
5. Desenvolupament de la 1 ^a versió de <i>RPPProjectManager</i>	200h
6. Elaboració de la memòria	150h
Total	900h

Taula 12: Temps estimat en la realització del projecte

20.2. Execució

Al següent calendari es pot observar quina ha sigut l'execució real del projecte des de que es va decidir el tema al Juliol de 2009 fins la realització de la memòria i la presentació del projecte, data en que podem donar per conclòs el projecte.

Al calendari, els dies amb els números vermells corresponen a dies festius i els que els tenen en negre corresponen a dies laborables. Per una altra banda, els dies amb fons de color:

- : corresponen a dies en que es va realitzar feina del PFC, però sense fixar un nombre d'hores determinat.
- : corresponen a dies en que es va realitzar feina del PFC, una mitja de 8 hores diàries.
- : corresponen a dies en que es va realitzar feina del PFC, una mitja de 5 hores diàries.
- : corresponen a dies en que no es va realitzar feina del PFC.

2009

Descripció 2009:

JULIOL						
DI	Dm	Dc	Dj	Dv	Ds	Dg
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

AGOST						
DI	Dm	Dc	Dj	Dv	Ds	Dg
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

SETEMBRE						
DI	Dm	Dc	Dj	Dv	Ds	Dg
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

OCTUBRE						
DI	Dm	Dc	Dj	Dv	Ds	Dg
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

NOVEMBRE						
DI	Dm	Dc	Dj	Dv	Ds	Dg
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

DESEMBRE						
DI	Dm	Dc	Dj	Dv	Ds	Dg
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Juliol, Agost

Lectures sobre Patrons de Requisits

Setembre

1-13: Lectures sobre Patrons de Requisits

14-15: Instal·lació de l'entorn de treball (16h)

16: Instal·lació i coneixement del funcionament de l'aplicació RPToolManager (8h)

17-30: Anàlisi de l'aplicació RPToolManager (84h)

Octubre

1-7: Correcció/Implementació dels errors de la capa de domini de RPToolManager (50h)

8-31: Correcció/Implementació dels errors de la capa de presentació (interfície) de RPToolManager (163h)

Novembre

1-14: Correcció/Implementació dels errors de la capa de presentació (interfície) de RPToolManager (100h)

15-16: Correcció/Implementació de les mancances de RPToolManager (13h)

17-24: Implementació de les noves funcionalitats de RPToolManager (58h)

24-30: Especificació de l'aplicació RPPProjectManager (42h)

Desembre

1-3: Implementació de la capa de domini i creació de la Base de Dades de RPPProjectManager (24h)

4-7: Implementació de la capa d'emmagatzematge de RPPProjectManager (26h)

8-12: Implementació de les funcionalitats de "Visualització de projectes i patrons" de RPPProjectManager (34h)

13-18: Implementació de les funcionalitats de "Construcció de projectes" de RPPProjectManager (48h)

19: Implementació de les funcionalitats de "Gestió de projectes" de RPPProjectManager (8h)

20-21: Implementació de les funcionalitats de "Interoperabilitat" de RPPProjectManager (13h)

22-31: Elaboració de la memòria (71h)

2010

Descripció 2010:

GENER						
DI	Dm	Dc	Dj	Dv	Ds	Dg
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Gener

1-17: Elaboració de la memòria (93h)

18: (Portar memòria a imprimir)

18-24: Preparació presentació

25: Presentació

Sabent ja les hores que s'han dedicat a cada activitat del projecte podem fer una comparació entre el nombre d'hores reals i el nombre d'hores estimades:

Activitat	Temps estimat	Temps real
1. Estat de l'art sobre Patrons de Requisits	100h	≈ 120h
2. Instal·lació de l'entorn de treball	100h	108h
3. Anàlisi de l'aplicació <i>RPToolManager</i>		
4. Desenvolupament de la 2 ^a versió de <i>RPToolManager</i>	350h	384h
5. Desenvolupament de la 1 ^a versió de <i>RPPProjectManager</i>	200h	195h
6. Elaboració de la memòria	150h	164h
Total	900h	971h

Taula 13: Comparació del temps estimat i el temps dedicat realment al projecte

21. Future Work

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
 20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

21. Future Work

En aquest projecte s'ha desenvolupat una primera versió de l'aplicació *RPPProjectManager* i s'ha realitzat una segona versió de *RPToolManager* lliure de bugs i amb noves funcionalitats. Tot i així, queda feina pendent en ambdues eines, especialment en *RPPProjectManager*, per tal d'aconseguir l'ús que es desitja d'elles (veure capítol 2.5 per més informació). A continuació s'explica, doncs, quina és aquesta feina pendent que queda com a treball futur per a noves versions de les aplicacions.

La primera funcionalitat a afegir en la següent versió de *RPPProjectManager* és la navegació seqüencial i automàtica pel catàleg de patrons de requisits. Tal com s'indica al mètode PABRE, la idea a grans trets és que durant el procés d'elicitació es recorri tot el catàleg de patrons i per cada patró, depenent de si interessa o no, instanciar-lo o no (per més informació sobre el mètode veure capítol 2.4). Ara mateix aquesta navegació s'ha de fer de forma manual. La idea és que en una segona versió, sigui l'aplicació qui faci aquesta navegació i et proposi el següent patró que toca considerar.

Per una altra banda, el feedback sobre l'ús del catàleg encara no es pot obtenir com seria desitjable a l'aplicació *RPPProjectManager*. Per aconseguir-ho, farien falta dues noves funcionalitats. La primera, l'aplicació hauria de permetre emmagatzemar, i també consultar, els motius pels quals s'ha afegit un requisit a un projecte que no instancia cap patró (ja sigui un requisit totalment nou o bé un requisit amb un patró associat). La segona, l'aplicació hauria de poder extreure una mena d'informe o similar on es poguessin visualitzar les estadístiques d'ús dels patrons dels requisits.

Un altre punt a tractar és com es produeix la comunicació entre les aplicacions *RPPProjectManager* i *RPToolManager*. Ara mateix, *RPPProjectManager* només guarda a la seva Base de Dades l'identificador del Pattern Item que un requisit de tipus "instància d'un patró" instancia i consulta de la Base de Dades de *RPToolManager* aquest Pattern Item quan ho necessita. El problema ve donat, entre d'altres, quan es modifica un Pattern Item amb l'aplicació *RPToolManager* i, per exemple, s'afegeix o s'elimina o es canvia de tipus un paràmetre. Això provocarà inconsistències a l'aplicació *RPPProjectManager* (si un requisit que instancia un patró té un valor de tipus enter que està associat a un paràmetre amb una mètrica tipus enter i canviem la mètrica d'aquest paràmetre per una de tipus set, a la Base de Dades de *RPPProjectManager* tindrem un valor tipus enter associat a un paràmetre tipus set!). I si eliminem directament un Pattern Item amb l'aplicació *RPToolManager* que és instanciat per un requisit d'un projecte a *RPPProjectManager*? És evident que tots aquests punts s'han de tractar si volem que es puguin realitzar modificacions al catàleg, i probablement noves versions de les dues aplicacions seran necessàries per aconseguir-ho.

A més, a l'aplicació *RPToolManager* faltaria implementar algun tipus de mètode que permetés indicar d'una manera unívoca, per una banda, la fórmula de correctesa dels paràmetres i, per una altra, la fórmula que indica si una extensió d'un patró es pot aplicar una o varies vegades o si de totes les extensions d'un patró només se'n pot instanciar una. Ara mateix, aquestes "fórmules" són explicades textualment a l'atribut *comentaris* d'un paràmetre o extensió, respectivament, però és necessari incorporar

algun mètode que permeti indicar aquestes fórmules de manera sistemàtica, per tal que després l'aplicació *RPPProjectManager* pugui tractar-les i fer que es compleixin durant el procés d'elicitació de requisits, feina que també queda pendent per futures versions d'aquesta eina.

Per últim, l'aplicació *RPPProjectManager* hauria de tractar les dependències entre els requisits que són instància d'un patró de requisit, perquè, per exemple, no es deixi afegir un requisit que instància un patró si contradiu a un altre requisit que també instància un patró dins del mateix projecte.

22. Conclusions

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software.....	239
20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
ANNEX 1: Modificacions al codi font de RPToolManager	265

22. Conclusions

Un cop acabat el projecte, i a punt de defensar-lo, puc fer ja una valoració global del què ha estat per a mi i les sensacions que he tingut al realitzar-lo. Com passa sempre que et trobes davant d'una situació desconeguda, un cop tenia ja el projecte escollit, la pregunta va ser "I ara, per on començo?" Però com tot, les coses no sempre són tant negres com es poden veure, i sent constant i prenent-se les coses seriosament, els problemes que en un principi et penses que tindràs i que no sabràs solucionar van desapareixent a mesura que es va treballant, i tot plegat es va transformant en un enriquiment personal constant.

La realització d'aquest projecte ha sigut possible gràcies als coneixements adquirits durant tots els anys d'universitat. Aquest coneixement no es basa només en els conceptes concrets que hem anat aprenent durant aquesta etapa, sense els quals, evidentment, aquest projecte avui no seria el que és, sinó també en la capacitat adquirida per aprendre ràpidament, buscar-te la vida per aconseguir una cosa que mai haves vist o realitzat, la manera d'enfrontar-te als problemes, la manera de solucionar-los i la manera d'adaptar-se al món actual de la informàtica, vesat al canvi constant.

El projecte final de carrera, apart de ser una bona oportunitat d'ampliar coneixements i aplicar els que ja es tenen, és una bona oportunitat per preparar-se de cara a la vida professional. Cal anar prenent decisions i contrastant-les amb el cap de projecte. Això m'ha fet aprendre a escoltar i a veure nous punts de vista que m'han enriquit tant professional com personalment.

El projecte ha estat, en sí mateix, la modificació d'un producte software ja existent i el desenvolupament d'un de nou en el que s'abasta totes les vessants de l'enginyeria del software. Això m'ha permès aplicar tot allò que vaig aprendre en les assignatures d'Enginyeria del Software 1 i 2, Base de Dades, etc. en un mateix projecte, obtenint així l'experiència mínima necessària per poder entrar en el món del desenvolupament software. Per altra banda, el fet de desenvolupar un projecte a partir d'una aplicació ja existent m'ha permès adquirir nous coneixements que em podran ser útils, ja que no tots els projectes comencen des de zero sinó que hi ha alguns que ja tenen una base tecnològica construïda.

Però si haig de destacar una cosa de tot el projecte, sense dubte, és el procés d'investigació que he hagut de realitzar per tal de poder realitzar el PFC. Això m'ha requerit consultar i agafar assiduïtat en la lectura de papers de caire científic en temes relacionats amb l'enginyeria del software, especialment en l'enginyeria de requeriments. I, també, m'ha permès poder entrar en contacte amb el món universitari de recerca i investigació, món que espero conèixer més a fons durant els pròxims anys de la meua vida professional.

23. Bibliografia

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2 ^a versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1 ^a versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1 ^a versió de l'aplic. RPProjectManager	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work.....	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD.....	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

23. Bibliografia

LLIBRES

[CFST00] Dolors Costal, Xavier Franch, M. Ribera Sancho, Ernest Teniente: “Enginyeria del Software. Especificació”. Edicions UPC, 2000.

[W07] Stephen Withall: “Software Requirement Patterns”. Microsoft Press, 2007.

[WS07] S. Warden, J. Shore: “The Art of Agile Development”. O'Reilly Media, 2007.

ARTICLES I TREBALLS

[RMFQ09] Samuel Renault; Óscar Mendez-Bonilla; Xavier Franch; Carme Quer. “PABRE: Pattern-based Requirements Elicitation”. In Third International Conference on Research Challenges in Information Science, 2009 (RCIS 2009).

[S05] Ian Somerville. “Integrated Requirements Engineering: A Tutorial”. In IEEE Software, 22 (1), pages 16-23, January/February 2005.

[MRRH05] Dan Matheson, Indrakshi Ray, Indrajit Ray, Siv Hilde Houmb. “Building Security Requirement Patterns for Increased Effectiveness Early in the Development Process” In SREIS Workshop 13th IEEE International Requirements Engineering Conference, Paris, France, August 2005.

[CJ07] Betty H.C. Cheng, Joanne M. Atlee. “Research Directions in Requirements Engineering”. In IEEE ICSE 2007, Future of Software Engineering 2007(FOSE '07).

[LMV97] W. Lam, J.A. McDermid, A.J. Vickers. “Ten Steps Towards Systematic Requirements Reuse”. In 3rd IEEE International Symposium on Requirements Engineering, January 5-8 1997 (RE'97), Annapolis, MD, USA.

[CR00] Jacob L. Cybulski, Karl Reed. “Requirements Classification and Reuse: Crossing Domain Boundaries”. In Proceedings of Software Reuse: Advances in Software Reusability, 6th International Conference, June 27-29 2000, (ICSR-6), Vienna, Austria.

[R96] Suzanne Robertson. “Requirement Patterns Via Events/Use Cases”. In The Atlantic Systems Guild Ltd., 1996.

[DBRT99] A. Durán Toro, B. Bernárdez Jiménez, A. Ruiz Cortés, M. Toro Bonilla. “A Requirements Elicitation Approach Based in Templates and Patterns”. In Workshop em Engenharia de Requisitos, Setembro 9-10 1999, (WER99), Buenos Aires, Argentina.

PÀGINES WEB

1. Metodologies i especificacions

- [WEB ISO 14764] Web ISO/IEC 14764 del 2006
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01703974&tag=1>
- [WEB Met. àgil] Metodologia Àgil
<http://www.agilemanifesto.org/>
- [WEB WhyAgileWorksFeedback] Perquè les metodologies àgils funcionen?
<http://www.ambysoft.com/essays/whyAgileWorksFeedback.html>
- [WEB Reqs. Volere] Anàlisi de Requeriments de Volere
<http://www.volere.co.uk/>
- [WEB FURPS+] Capturing Architectural Requirements, The FURPS+ System for Classifying Requirements
<http://www.ibm.com/developerworks/rational/library/4706.html>
- [WEB UML2.2] Especificació UML 2.2
<http://www.omg.org/spec/UML/2.2/>

2. Entorns de Desenvolupament

- [WEB Eclipse] Eclipse
<http://www.eclipse.org>
- [WEB Java] Java
<http://www.java.sun.com>
- [WEB Hibernate] Hibernate
<http://www.hibernate.org/>
- [WEB SQuirreL] SQuirreL-SQL CLient
<http://squirrel-sql.sourceforge.net/>

3. APIs

- [WEB Java Api] Java J2SE 5.0 API
<http://java.sun.com/j2se/1.5.0/docs/api/>

- [WEB Eclipse API] Eclipse API
<http://help.eclipse.org/help31/index.jsp?topic=/org.eclipse.platform.doc.isv/reference/api/overview-summary.html>

4. Tutorials i Manuals

- [WEB W3C] W3C Schools
<http://www.w3schools.com>
- [WEB Eclipse Help] Eclipse Help
<http://help.eclipse.org/help32/index.jsp>
- [WEB Wiki Hibernate] Wiki Hibernate
<https://www.hibernate.org/37.html>

24. Contingut del CD

1. Introducció	11
2. Patrons de requisits.....	15
3. Eines de Treball.....	29
4. Arquitectura: En tres capes	35
 PART 1: Desenvolupament de la 2ª versió de l'aplic. RPToolManager	39
5. Manteniment i Extensió de Software	41
6. Anàlisi de la 1ª versió de RPToolManager	47
7. Model Conceptual	61
8. Base de Dades.....	67
9. Segona versió de RPToolManager	75
 PART 2: Desenvolupament de la 1ª versió de l'aplic. RPProjectManager	79
10. Metodologia: Àgil	81
11. Anàlisi de Requeriments	85
12. Model de Casos d'ús.....	97
13. Model Conceptual	125
14. Model de Comportament	133
15. Disseny de la Interfície	167
16. Reutilització de Components	189
17. Base de Dades.....	195
18. Document de Disseny.....	201
19. Proves del Software	239
 20. Planificació i Execució del Projecte	245
21. Future Work	251
22. Conclusions.....	255
23. Bibliografia	257
24. Contingut del CD	261
 ANNEX 1: Modificacions al codi font de RPToolManager	265

24. Contingut del CD

Dins el CD que es proporciona amb aquesta memòria podem trobar:

- **readme.txt.** Aquest arxiu conté les instruccions de com executar les versions desenvolupades en aquest projecte de les aplicacions *RPToolManager* i *RPPProjectManager*.
- **Executables.** Aquest directori conté tots els arxius necessaris per executar les versions desenvolupades en aquest projecte de les aplicacions *RPToolManager* i *RPPProjectManager*. Dins d'aquest directori trobem:
 - **database.** Directori que conté la Base de Dades del repositori de patrons.
 - **databaseProjects.** Directori que conté la Base de Dades del repositori de projectes.
 - **images.** Directori que conté les imatges que s'utilitzen durant l'execució de les aplicacions *RPToolManager* i *RPPProjectManager*.
 - **debug.log** i **derby.log.** Arxius de log utilitzats per les aplicacions.
 - **DerbyServerProjects.jar.** Arxiu executable que s'utilitza per obrir la connexió amb les Bases de Dades, tant de projectes com de patrons.
 - **log4j.cfg.** Arxiu que conté la configuració de en quin fitxer escriure els logs.
 - **RPPProjectManager.jar.** Arxiu executable de l'eina *RPPProjectManager*.
 - **RPToolManager.jar.** Arxiu executable de l'eina *RPToolManager*.

ANNEX

ANNEX 1: Modificacions al codi font de RPToolManager

A continuació es mostren les modificacions que s'han realitzat al codi font de l'aplicació *RPToolManager* per tal de corregir i incorporar els errors, mancances i noves funcionalitats detectades durant l'anàlisi de l'aplicació (per més informació sobre l'anàlisi consultar el capítol 6).

Errors a nivell de domini

Error		Solució
D1	En un requirement form, es permet afegir més d'una extension amb el mateix nom.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>RequirementForm.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> per tal que abans d'afegir un nou extended part a un patró, es comprovi primer si hi ha cap altre extended part dins el patró amb el mateix nom. S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que capturi i mostri l'error quan s'intenta afegir un nou extended part a un patró amb un nom igual al d'un altre extended del mateix patró.
D2	Dins d'un classification schema, es pot crear més d'un root classifier amb el mateix nom.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>ClassificationSchema.java</i> del paquet <i>edu.upc.gessi.rptool.domain.schema</i> per tal que abans d'afegir un nou root classifier a un classification schema, es comprovi primer si hi ha cap altre root classifier dins el classification schema amb el mateix nom. S'ha modificat del fitxer <i>MediatorSchemas.java</i> del paquet <i>edu.upc.gessi.rptool.storage</i> el mètode <i>save(RootClassifier c, ClassificationSchema parent)</i> per tal que capturi l'error que es pot produir ara a l'afegir un root classifier a un classification schema.

Annex 1: Modificacions al codi font de RPToolManager

D3	Dins d'un root classifier o d'un decomposed classifier, es pot crear més d'un basic/decomposed classifier amb el mateix nom.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>InternalClassifier.java</i> del paquet <i>edu.upc.gessi.rptool.domain.schema</i> per tal que abans d'afegir un nou internal classifier a un classifier, es comprovi primer si hi ha cap altre internal classifier dins del classifier amb el mateix nom. S'ha modificat del fitxer <i>MediatorSchemas.java</i> del paquet <i>edu.upc.gessi.rptool.storage</i> el mètode <i>save(InternalClassifier c, InternalClassifier parent)</i> per tal que capturi l'error que es pot produir ara a l'afegir un internal classifier a un classifier.
D4	Quan estàs creant/modificant una mètrica de tipus string, deixa afegir com a default value un string buit o un string que sigui tot espais en blanc.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>StringMetric.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> el mètode <i>setDefaultValue(..)</i> per tal que comprovi que si hi ha default value, aquest no és un valor buit o ple d'espais en blanc. S'ha modificat del fitxer <i>DialogStringMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi l'error.
D5	Quan estàs creant/modificant una mètrica de tipus integer, deixa afegir valors al mínim i al màxim que no compleixen la restricció de que el mínim ha de ser menor o igual que el màxim.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricsErrors.java</i> al paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per afegir el mètode <i>showError(...)</i> que mostra la finestra amb el missatge d'error comentat. S'ha modificat la classe <i>ValueException.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal que aquest tipus d'error incorpori també un missatge.
D6	Quan estàs creant/modificant una mètrica de tipus integer, deixa afegir valors al default value que no compleixen la restricció de que aquest valor ha d'estar comprès entre el mínim i el màxim.	
		<ul style="list-style-type: none"> S'ha modificat el fitxer <i>IntegerMetric.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal d'incorporar els missatges als errors de tipus <i>ValueException</i>. S'ha modificat del fitxer <i>DialogIntegerMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi els errors.

D7	Quan estàs creant/modificant una mètrica de tipus float, deixa afegir valors al mínim i al màxim que no compleixen la restricció de que el mínim ha de ser menor o igual que el màxim.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>FloatMetric.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal d'incorporar els missatges als errors de tipus <i>ValueException</i>.• S'ha modificat del fitxer <i>DialogFloatMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi els errors.
D8	Quan estàs creant/modificant una mètrica de tipus float, deixa afegir valors al default value que no compleixen la restricció de que aquest valor ha d'estar comprès entre el mínim i el màxim.	

Errors a nivell d'interfície

Error		Solució
I1	A la finestra de patrons i de mètriques, no funciona el botó per minimitzar la finestra.	<ul style="list-style-type: none"> De fet, no es que el botó per minimitzar no funcioni, sinó que el contenidor on apareix la finestra de mètriques i la de patrons obliga a aquestes a que sempre tinguin la mateixa mida que el contenidor. Per evitar confusions, s'ha eliminat el botó de minimitzar i s'ha permès al botons que tornen a obrir les finestres de mètriques i patrons que puguin obrir i tancar aquestes finestres, de manera que ara l'opció de minimitzar/restaurar les finestres a passat als botons de la barra de menús. S'ha modificat el fitxer <i>InternalFrameGeneric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que no mostri el botó de minimitzar. S'ha modificat el fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal d'incorporar els mètodes <i>getInstances(...)</i> i <i>unsetInstances(...)</i> necessaris per controlar el botó que minimitza/restaura la finestra de patrons. S'ha modificat el fitxer <i>MetricssInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal d'incorporar els mètodes <i>getInstance(...)</i> i <i>unsetInstance(...)</i> necessaris per controlar el botó que minimitza/restaura la finestra de mètriques. S'ha modificat el fitxer <i>MainFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.mainFrame</i> per tal de modificar els mètodes que controlen les accions dels botons que minimitzen/restauren la finestra de patrons i la de mètriques.
I2	A la finestra de patrons i de mètriques, quan el nom del patró o de la mètrica, respectivament, és molt llarg no es veu sencer perquè no apareix l'scrollbar horitzontal; s'ha de modificar la mida de la finestra manualment per tal de veure el nom sencer.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>InternalFrameGeneric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que aparegui quan sigui necessari l'scrollbar horitzontal .

Annex 1: Modificacions al codi font de RPToolManager

I3	A la finestra de patrons, al desplegable per seleccionar el classification schema no es veuen sencers els noms llargs.	<ul style="list-style-type: none"> Solució: quan el nom sigui massa llarg, truncar-lo i posar punts suspensius. Per aconseguir-ho, s'han hagut de realitzar les següents modificacions al fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i>: <ul style="list-style-type: none"> S'ha afegit un nou atribut a la classe, <i>schemasDots</i>, per tal de guardar els noms dels classifications schemas truncats i no perdre els noms originals, que es continuen guardant a l'atribut <i>schemas</i>. S'han afegit les funcions <i>addDots(Vector v)</i> i <i>addDots(String s)</i> per si, en cas de que algun dels classification schemas té més de 50 caràcters, truncar-lo i afegir-li punts suspensius.
I4	A la finestra de patrons, en l'esquema <i>Alphabetical list</i> els patrons no apareixen ordenats alfabèticament.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> el mètode <i>loadPatternsAlphabetically()</i> per tal que ordeni els patrons alfabèticament.
I5	A la finestra de patrons, a l'esborrar un classification schema ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del classification schema esborrat.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> el mètode <i>removeClassifier(...)</i> per tal que s'actualitzin tant la finestra taronja com l'arbre.
I7	A la finestra de patrons, a l'esborrar un root classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del root classifier esborrat.	
I9	A la finestra de patrons, a l'esborrar un basic/decomposed classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació del basic/decomposed classifier esborrat.	

Annex 1: Modificacions al codi font de RPToolManager

I6	A la finestra de patrons, al modificar un root classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació d'abans de la modificació.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> l'acció de l'opció <i>ModifyClassifier...</i> per a que si el node és un root classifier o un classifier s'actualitzin tant la finestra taronja com l'arbre.
I8	A la finestra de patrons, al modificar un basic/decomposed classifier ni l'arbre de navegació ni la finestra taronja de sota s'actualitzen i persisteix la informació d'abans de la modificació.	
I10	A la finestra de patrons, a l'esborrar un patró la informació del mateix persisteix a la part inferior taronja de la finestra tot i que el patró ja ha sigut esborrat.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> el mètode <i>jTree1_valueChanged(...)</i> per a que si el node que s'ha canviat és null (perquè s'ha esborrat) esborri el contingut de la finestra taronja dels patrons i mostri un altre missatge.
I11	A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb un nom que és un string buit o un conjunt d'espais en blanc, no deixa guardar però tampoc es mostra cap missatge d'error.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>DialogSchema.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogSchemas</i> el mètode <i>okAction()</i> per tal que capturi l'error.
I12	A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom del classification schema. Per això: <ul style="list-style-type: none"> S'ha creat la nova classe <i>DocumentWithLimitCharacters.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que un <i>JTextField</i> no pugui contenir més de 250 caràcters. S'ha modificat el fitxer <i>GenericObjectTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal de limitar amb la classe anterior el nombre màxim de caràcters al <i>JTextField</i> on s'introdueix el nom.

Annex 1: Modificacions al codi font de RPToolManager

I13	A la finestra d'edició de classification schemas, quan es vol guardar un classification schema amb el nom d'un classification schema que ja existeix, no deixa guardar però tampoc es mostra cap missatge d'error.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogSchema.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogSchemas</i> el mètode <i>okAction()</i> per tal que capturi l'error.
I14	A la finestra d'edició de classifiers, quan es vol guardar un classifier (ja sigui root, decomposed o basic) amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom del classifier. Per això: <ul style="list-style-type: none"> S'ha creat la nova classe <i>DocumentWithLimitCharacters.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que un <i>TextField</i> no pugui contenir més de 250 caràcters. S'ha modificat el fitxer <i>GenericObjectTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal de limitar amb la classe anterior el nombre màxim de caràcters al <i>TextField</i> on s'introdueix el nom.
I15	A la finestra d'edició de root classifiers, la finestra té com a títol "classifier" en comptes de "root classifier".	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogClassifier.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogSchemas</i> per corregir l'error explicat.
I16	A la finestra d'edició de patrons, quan es vol guardar un patró amb un nom que és un string buit o una cadena d'espais en blanc, la finestra que es mostra amb l'error no és gaire aclaridora.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de clarificar l'error esmentat.
I17	A la finestra d'edició de patrons, quan es vol guardar un patró amb el nom d'un altre patró que ja existeix, la finestra que es mostra amb l'error no és gaire aclaridora.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de clarificar l'error esmentat.

Annex 1: Modificacions al codi font de RPToolManager

I18	A la finestra d'edició de patrons, quan es vol guardar un patró amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom del patró. Per això: <ul style="list-style-type: none"> S'ha creat la nova classe <i>DocumentWithLimitCharacters.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que un <i>TextField</i> no pugui contenir més de 250 caràcters. S'ha modificat el fitxer <i>GenericObjectTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal de limitar amb la classe anterior el nombre màxim de caràcters al <i>TextField</i> on s'introdueix el nom.
I19	A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp <i>author</i> és massa llarg, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el camp <i>author</i> del patró. Per això: <ul style="list-style-type: none"> S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de limitar amb la classe <i>DocumentWithLimitCharacters</i> el nombre màxim de caràcters al <i>TextField</i> on s'introdueix l'autor del patró.
I20	A la finestra d'edició de patrons, a la pestanya <i>attributes</i> del patró, al calendari del camp <i>version</i> , a la capsa de text per introduir l'any, si introduïm anys més grans de 9999, tot i ser anys no vàlids per l'aplicació, surten escrits en color verd volent dir que són vàlids.	<ul style="list-style-type: none"> S'ha afegit al fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> el codi que estableix l'any més gran que es pot introduir al camp <i>year</i> del calendari.
I21	A la finestra d'edició de patrons, a la pestanya <i>dependencies</i> , es pot afegir la mateixa <i>dependency</i> més d'un cop, tot i que després no es deixa guardar els canvis.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que es mostri un error quan això succeeix.

Annex 1: Modificacions al codi font de RPToolManager

I22	A la finestra d'edició de patrons, quan es vol guardar un patró que té dues dependències iguals, el missatge d'error que apareix és incorrecte (surt el mateix missatge d'error que quan el nom del patró ja existeix).	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que de modificar el missatge d'error.
I23	A la finestra per seleccionar sources (a la qual s'accedeix des de la finestra d'edició de patrons o de mètriques quan es vol afegir una nova source), el missatge d'error que es mostra quan s'intenta esborrar una source que està associada a un patró o a una mètrica és incorrecte.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSourceList.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal que el missatge d'error esmentat sigui més precís.
I24	A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), no es mostra cap error quan l'identificador de la source és massa llarg com per guardar-lo a la Base de Dades.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSource.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal de mostrar l'error comentat.
I25	A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), no es mostra cap error quan l'identificador de la source ja existeix.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSource.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal de mostrar l'error comentat.

Annex 1: Modificacions al codi font de RPToolManager

I26	A la finestra per editar sources (a la qual s'accedeix des de la finestra per seleccionar sources), si vols cancel·lar abans d'haver introduït un identificador per la source no et deixa, et demana que introdueixis un identificador.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSource.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal de corregir l'error comentat.
I27	A la finestra per seleccionar keywords (a la qual s'accedeix des de la finestra d'edició de patrons quan es vol afegir una nova keyword), el missatge d'error que es mostra quan s'intenta esborrar una keyword que està associada a un patró és incorrecte.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogKeywordsList.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal que el missatge d'error comentat sigui correcte.
I28	A la finestra per editar keywords (a la qual s'accedeix des de la finestra per seleccionar keywords), es deixa introduir noves keywords que siguin només espais en blancs o modificar una keyword a només espais en blancs.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSourceList.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per corregir l'error esmentat.
I29	A la finestra per editar keywords (a la qual s'accedeix des de la finestra per seleccionar keywords), el missatge que surt quan una keyword és massa llarga per guardar-la a la Base de Dades o quan la keyword ja existeix és el mateix; estaria bé diferenciar-los.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogSourceList.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal de diferenciar els errors esmentats.

Annex 1: Modificacions al codi font de RPToolManager

I30	A la finestra d'edició de patrons, quan en el formulari que surt quan es prem el botó <i>New Form...</i> s'introdueix un string buit o un seguit d'espais en blanc l'aplicació no crea el nou form però tampoc mostra cap error.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que es mostri l'error produït.
I31	A la finestra d'edició de patrons, es pot modificar el nom d'un form per el d'un altre form que ja existeix en el patró o canviar-lo a un string buit i es pot guardar sense cap problema.	<ul style="list-style-type: none"> S'ha afegit al fitxer <i>RequirementPattern.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> el mètode <i>testNameForm(...)</i> que comprova si existeix cap altre form al patró amb el nom passat com a paràmetre. S'ha afegit al fitxer <i>GenericObjectTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> el mètode <i>addNameFocusListener(...)</i> per tal de poder afegir un focus listener al JTextField del nom. S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de modificar el JTextField del nom per tal que si es canvia el nom a un string buit o a un nom d'un altre form salti un error i proposi un nou nom.
I32	A la finestra d'edició de patrons, quan es vol guardar un patró que té un requirement form amb un nom massa llarg, i per tant no deixa guardar-lo a la base de dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom form. Per això: <ul style="list-style-type: none"> S'ha modificat el fitxer <i>GenericObjectTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal de limitar amb la classe <i>DocumentWithLimitCharacters</i> el nombre màxim de caràcters al JTextField on s'introdueix el nom. S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que al prémer el botó "New form..." no deixi introduir un nom de form amb més de 250 caràcters.

Annex 1: Modificacions al codi font de RPToolManager

I33	A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp <i>author</i> d'un dels seus requirement forms és massa llarg, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none">• Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el camp <i>author</i> del form. Per això:<ul style="list-style-type: none">◦ S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de limitar amb la classe <i>DocumentWithLimitCharacters</i> el nombre màxim de caràcters al JTextField on s'introdueix l'autor del form.
I34	A la finestra d'edició de patrons, quan es vol guardar un patró tal que el valor del camp <i>form text</i> d'una de les seves fixed/extended part d'un dels seus requirement forms és buit, la finestra que es mostra amb l'error no és gaire aclaridor.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de modificar el missatge d'error esmentat.
I35	A la finestra d'edició de patrons, quan es crea un nou requirement form i al form text de la fixed part es posen només espais en blancs o si quan ja està creat es modifica aquest form text a un string buit o a un conjunt d'espais, deixa guardar el patró i això és incorrecte.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de corregir l'error esmentat.

I36	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan es crea una nova extended part dins d'un form i al seu form text es posen només espais en blancs o si quan ja està creat es modifica aquest form text a un string buit o a un conjunt d'espais, deixa guardar el patró i això és incorrecte.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de corregir l'error esmentat.
I37	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan el nom d'una extension és més gran del que cap al desplegable per seleccionar l'extended part, el nom no apareix sencer i el desplegable acaba ocupant tota la finestra, fent desaparèixer la part de navigation de la finestra d'edició de patrons.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>ExtendedPattern.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> per tal que la funció <i>toString(...)</i> retorni el nom de l'extensió, però que si aquesta té més de 55 caràcters, retorni només els primers 55 caràcters amb uns punts suspensius al final. S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que el desplegable no es pugui fer mai més gran que una mida prefixada
I38	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan s'intenta afegir un nou extended part amb un nom buit, no s'afegeix l'extended part però no es mostra cap error com que no s'ha afegit.	<ul style="list-style-type: none"> Modificar el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que mostri un missatge quan es produeix l'error comentat.

Annex 1: Modificacions al codi font de RPToolManager

I39	A la finestra d'edició de patrons, quan es vol guardar un patró que té una extended part amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom de l'extension. Per això: <ul style="list-style-type: none"> S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que al prémer el botó "New" a la pestanya Extension no deixi introduir un nom amb més de 250 caràcters.
I40	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan s'esborra una extended part, si aquesta era l'última que quedava, es queden les seves dades als camps del formulari, tot i que l'extension s'ha eliminat.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> l'acció del botó "Remove..." per tal que si l'extensió que s'esborra era la última que quedava, es buidin tots els camps de la pestanya Extensions
I41	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts i es vol afegir un nou paràmetre, el sistema deixa que introduïm un paràmetre sense nom.	<ul style="list-style-type: none"> Afegir al fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> la comprovació al nom per a que no accepti com a nom vàlid ni un string buit ni un conjunt d'espais en blanc.
I42	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts i es vol afegir un nou paràmetre, el missatge d'error que surt quan el nom del paràmetre ja existeix en la fixed o extended part no és gaire aclaridor.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de modificar els missatges d'error esmentats.

Annex 1: Modificacions al codi font de RPToolManager

I43	A la finestra d'edició de patrons, l'aplicació permet que quan estàs modificant la informació d'un paràmetre, es canviï el nom del paràmetre al d'un altre paràmetre que ja existeix dins de la mateixa fixed/extended part, o que es canviï el nom a un string buit o a un conjunt d'espais en blanc.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>ParameterTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal d'afegir dos nous atributs a aquesta classe que ens permetin saber a quina fixed o extended part pertany el paràmetre que s'està editant. Això ha implicat també modificar les constructores per poder assignar valors als nous atributs i modificar el JTextField del nom per tal que si es canvia el nom a un string buit o a un nom d'un altre paràmetre salti un error i proposi un nou nom correcte per aquesta fixed o extended part concreta. • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal d'actualitzar i donar valor als nous atributs de la classe anterior.
I44	A la finestra d'edició de patrons, quan es vol guardar un patró que té un paràmetre amb un nom massa llarg, i per tant no deixa guardar-lo a la Base de Dades, la finestra que es mostra amb l'error no explica quin és l'error que ha ocorregut.	<ul style="list-style-type: none"> • Per tal de solucionar aquest problema s'ha pres l'opció de no deixar que l'usuari introdueixi més de 250 caràcters en el nom del paràmetre. Per això: <ul style="list-style-type: none"> ○ S'ha modificat el fitxer <i>ParameterTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de limitar amb la classe <i>DocumentWithLimitCharacters</i> el nombre màxim de caràcters al JTextField on s'introdueix el nom. ○ S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que al prémer el botó "New parameter..." no deixi introduir un nom de paràmetre amb més de 250 caràcters.
I45	A la finestra d'edició de patrons, quan es vol guardar un patró que té un paràmetre sense mètrica assignada, el missatge d'error que es mostra no és gaire aclaridor.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de corregir el missatge d'error esmentat.
I46	Cada cop que li donem al botó per obrir i tancar la finestra de mètriques, aquesta surt en una posició diferent.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>GenericMainFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que a l'obrir una finestra sempre es mostri a la mateixa posició.

Annex 1: Modificacions al codi font de RPToolManager

I47	A la finestra de mètriques, quan deixem el cursor a sobre d'un element de l'arbre surt al quadradet groc típic però té escrit una text incorrecte.	<ul style="list-style-type: none">• S'ha decidit eliminar el ToolTip de la finestra de mètriques per dos motius: el primer, la finestra de patrons no té cap ToolTip associat, i el segon, el ToolTip en l'arbre que mostra les mètriques no crec que pugui aportar cap informació útil. Per això, s'ha modificat el fitxer <i>GenericTreeCellRenderer.java</i> del paquet <i>edu.upc.gessi.rptool.gui.common</i> per tal que no mostri el ToolTip en l'arbre de mètriques.
I48	A la finestra de mètriques, a la informació que surt a la part inferior taronja de la finestra sobre la mètrica que hi ha seleccionada, si el nom de la mètrica conté el caràcter '<', només surt el nom fins al caràcter '<'. Ídem pel caràcter '>'.	<ul style="list-style-type: none">• S'ha modificat el fitxer <i>MetricsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal d'evitar l'error que es comenta, substituint els caràcters '<' o '>' que qualsevol camp d'una mètrica podia contenir pel seu equivalent en HTML (&lt; o &gt;, respectivament).
I49	A la finestra de mètriques, si fas clic dret a sobre d'una mètrica de tipus time point no funciona l'opció del menú emergent <i>Modify...</i> , no fa res.	<ul style="list-style-type: none">• S'ha modificat el codi del fitxer <i>MetricsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal que l'opció del menú emergent quan es modificava una mètrica de tipus time point funcionés.• També s'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que s'omplís el diàleg que surt per modificar la mètrica amb els valors que aquesta ja tenia.

I50	A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que no hi ha nom a la mètrica, però tampoc et deixa guardar.	<ul style="list-style-type: none"> S'han modificat el mètode <i>okAction()</i> dels fitxers <i>DialogStringMetric.java</i>, <i>DialogIntegerMetric.java</i>, <i>DialogFloatMetric.java</i>, <i>DialogTimePointMetric.java</i>, <i>DialogDomainMetric.java</i> i <i>DialogSetMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que capturi i mostri els errors comentats. S'ha creat el nou fitxer <i>MetricsErrors.java</i> al paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> i s'han afegit els mètodes <i>duplicateKeyValueTypeError(...)</i>, <i>metricNameTooLongError(...)</i> i <i>showError(...)</i> que mostren les finestres amb els missatges d'error dels errors comentats.
I51	A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que el nom de la mètrica és massa llarg, però tampoc et deixa guardar.	
I52	A la finestra d'edició de mètriques, sigui del tipus que sigui aquesta, no et mostra l'error de que ja existeix una mètrica amb el mateix nom que aquesta, però tampoc et deixa guardar.	
I53	A la finestra d'edició de mètriques de tipus integer, si estàs modificant una mètrica, l'aplicació no et deixa guardar els canvis tot i que siguin valors correctes. Per exemple, si es vol modificar un <i>integer</i> que s'ha creat amb mínim -12 i màxim -11 i es vol modificar i posar mínim 30 i màxim 31, tota l'estona dona error (si es crea una nova mètrica amb aquests mínims i màxims si que la crea bé).	<ul style="list-style-type: none"> El problema era que quan es volia modificar una mètrica de tipus integer, sempre es comprovava que el mínim fos més petit o igual que el màxim. Llavors si teníem inicialment un integer amb mínim -12 i màxim -11, i volíem canviar als valors 30 i 31, respectivament, com primer es modificava el mínim posant-lo a 30 i encara no s'havia modificat el màxim, s'incomplia la restricció de que el mínim és més petit que el màxim, ja que 30 no és més petit o igual que -11. Canviar al fitxer <i>DialogIntegerMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> la forma de modificar una mètrica de tipus integer per tal que permeti fer la modificació que a l'error es comenta.

I54	A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que no hi ha cap mínim assignat o de que aquest no té format numèric d'enter, però tampoc et deixa guardar.	
I55	A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que no hi ha cap màxim assignat o de que aquest no té format numèric d'enter, però tampoc et deixa guardar.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricsErrors.java</i> al paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per afegir els mètodes <i>integerFloatMetricMinimumValueNotNumberError(...)</i>, <i>integerFloatMetricMaximumValueNotNumberError(...)</i> i <i>integerFloatMetricDefaultValueNotNumberError(...)</i> que mostren les finestres amb els missatges d'error comentats. S'ha modificat del fitxer <i>DialogIntegerMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi els errors.
I56	A la finestra d'edició de mètriques de tipus integer, no et mostra l'error de que quan estant seleccionada la casella per introduir el default value no hi ha cap default value assignat o que el default value introduït no té format numèric d'enter, però tampoc et deixa guardar.	
I57	A la finestra d'edició de mètriques de tipus float, si estàs modificant una mètrica, l'aplicació no et deixa guardar els canvis tot i que siguin valors correctes. Per exemple, si es vol modificar un <i>float</i> que s'ha creat amb mínim -12 i màxim -11 i es vol modificar i posar mínim 12 i màxim 16, tota l'estona dona error (si es crea una nova mètrica amb aquests mínims i màxims si que la crea bé).	<ul style="list-style-type: none"> El problema era que quan es volia modificar una mètrica de tipus float, sempre es comprovava que el mínim fos més petit o igual que el màxim. Llavors si teníem inicialment un float amb mínim -12 i màxim -11, i volíem canviar als valors 30 i 31, respectivament, com primer es modificava el mínim posant-lo a 30 i encara no s'havia modificat el màxim, s'incomplia la restricció de que el mínim és més petit que el màxim, ja que 30 no és més petit o igual que -11. Canviar al fitxer <i>DialogFloatMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> la forma de modificar una mètrica de tipus float per tal que permeti fer la modificació que a l'error es comenta.

I58	A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que no hi ha cap mínim assignat o de que aquest no té format numèric de float, però tampoc et deixa guardar.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>DialogFloatMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi els errors.
I59	A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que no hi ha cap màxim assignat o de que aquest no té format numèric de float, però tampoc et deixa guardar.	
I60	A la finestra d'edició de mètriques de tipus float, no et mostra l'error de que quan estant seleccionada la casella per introduir el default value no hi ha cap default value assignat o que quan el default value introduït no té format numèric de float, però tampoc et deixa guardar.	
I61	A la finestra d'edició de mètriques de tipus string, no et mostra l'error quan es viola la condició de longitud màxima del default value, però tampoc et deixa guardar.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricsErrors.java</i> al paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per afegir el mètode <i>stringMetricDefaultValueTooLongError(...)</i> que mostra la finestra amb el missatge d'error comentat. S'ha modificat del fitxer <i>DialogStringMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi l'error.
I62	A la finestra d'edició de mètriques de tipus time point, el formulari té com a títol "new string metric".	<ul style="list-style-type: none"> Modificar el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal de canviar el títol dels formularis de nova mètrica i modificació de mètriques de tipus time point.

Annex 1: Modificacions al codi font de RPToolManager

I63	A la finestra d'edició de mètriques de tipus time point, el camp <i>day</i> del formulari deixa introduir com a màxim fins el dia 31, cosa que està bé, però no tots els mesos tenen 31 dies. S'hauria de comprovar això quan s'introdueix també un mes.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que es comprovi, si el camp mes i dia estan seleccionats, si el valor del dia és correcte pel valor del mes.
I64	A la finestra d'edició de mètriques de tipus time point, si s'introdueix al camp <i>hour</i> o <i>minute</i> un nombre superior al màxim permès, en lloc de substituir-se pel nombre més gran (23 o 59), es substitueix per un nombre que no té ni cap ni peus que sigui aquell nombre. Passa el mateix amb el camp <i>day</i> i <i>second</i> .	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que quan s'introdueix un valor incorrecte en els camps <i>Day</i>, <i>Hour</i>, <i>Minute</i> o <i>Second</i> es mostri un missatge indicant de l'error i el camp en qüestió s'estableixi amb el valor mínim que pot tenir.
I65	A la finestra d'edició de mètriques de tipus domain, quan s'està modificant una mètrica, no deixa guardar els canvis mai, encara que tots els valors introduïts siguin correctes.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>DialogDomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que quan es modifica una mètrica d'aquest tipus s'utilitzi la funció adequada per actualitzar aquest tipus de mètriques (abans s'utilitzava la funció genèrica, que no funcionava bé per aquest tipus).
I66	A la finestra d'edició de mètriques de tipus domain, deixa guardar la mètrica tot i que no hi hagi cap value afegit, i això no hauria de ser possible.	<ul style="list-style-type: none"> Canviar al fitxer <i>DialogDomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> l'acció que realitza el botó "Ok" per tal que comprovi que almenys hi ha un valor assignat.
I67	A la finestra d'edició de mètriques de tipus domain, deixa afegir valors buits, i això no hauria de ser possible.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogDomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que es comprovi a l'accionar el botó "Add" que el valor no és un string buit ni un conjunt d'espais en blanc.

Annex 1: Modificacions al codi font de RPToolManager

I68	A la finestra d'edició de mètriques de tipus domain, deixa afegir valors molt llargs, i això no hauria de ser possible ja que després quan vols guardar la mètrica no et deixa guardar-la però tampoc mostra l'error que es produeix.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricsErrors.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per afegir el mètode <i>metricFieldTooLongError(...)</i> i el mètode <i>valueTooLong(..)</i> que comproven si es produeix l'error comentat i mostra la finestra amb el missatge d'error corresponent. S'ha modificat del fitxer <i>DialogDomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi l'error.
I69	A la finestra d'edició de mètriques de tipus domain, es deixa afegir dos o més valors possibles iguals, però després no et deixa guardar i l'error que et mostra només diu que hi ha algun camp incorrecte.	<ul style="list-style-type: none"> S'ha modificat la classe <i>PossibleValueException.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal que aquest tipus d'error incorpori també un missatge. S'ha modificat el fitxer <i>DomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal d'incorporar els missatges als errors de tipus <i>PossibleValueException</i> i <i>ValueException</i>. S'ha modificat el fitxer <i>MetricsErrors.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per afegir el mètode <i>showError(...)</i> que mostra l'error corresponent de l'excepció de tipus <i>PossibleValueException</i>. S'ha modificat del fitxer <i>DialogDomainMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi l'error.
I70	A la finestra d'edició de mètriques de tipus set, no et mostra l'error quan no hi ha cap mètrica simple assignada, però tampoc et deixa guardar.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>DialogSetMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal que capturi i mostri l'error.
I71	No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació: <ol style="list-style-type: none"> 1. Estem editant un patró. 2. Afegim una nova dependency amb qualsevol patró. 3. Esborrem el patró de la dependency afegida. 4. Intentem guardar. 	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de mostrar un missatge d'error quan això succeeix.

I72	<p>No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:</p> <ol style="list-style-type: none"> 1. Estem editant un patró. 2. Afegim un nou paràmetre a qualsevol fixed o extended part d'un dels seus requirement forms i li assignem una mètrica qualsevol. 3. Esborrem la mètrica que hem assignat al paràmetre. 4. Intentem guardar. 	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de mostrar un missatge d'error quan això succeeix.
I73	<p>No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:</p> <ol style="list-style-type: none"> 1. Estem modificant un patró. 2. L'esborrem. 3. Intentem guardar. 	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de mostrar un missatge d'error quan això succeeix.
I74	<p>No es mostra cap missatge d'error ni deixa guardar quan ocorre la següent situació:</p> <ol style="list-style-type: none"> 1. Estem editant una mètrica de tipus set. 2. Assignem qualsevol mètrica simple com a tipus del set. 3. Esborrem la mètrica que hem assignat al tipus del set. 4. Intentem guardar. 	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>DialogSetMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode <i>okAction()</i> per tal de mostrar un missatge d'error quan això succeeix.

Mancances

Mancança		Solució
M1	Faria falta un botó similar al que obre la finestra de mètriques per obrir la finestra de patrons, perquè si es tanca aquesta finestra no hi ha manera de tornar-la a obrir sense tancar l'aplicació i tornar-la a executar.	<ul style="list-style-type: none"> Afegir al fitxer <i>MainFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.mainFrame</i> el codi necessari per tornar a obrir la finestra de patrons un cop s'ha tancada.
M2	A la finestra de patrons, al crear un nou classification schema, el desplegable del classification schema seleccionat no canvia al nou ítem creat.	<ul style="list-style-type: none"> S'ha modificat del fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> l'acció del botó “New” per tal que després d'inserir el nou classification schema, el desplegable canviï al nou ítem i s'actualitzi la finestra taronja
M3	A la finestra de patrons, al crear un nou classification schema, el desplegable que conté tots els classifications schema es desordena. A més, l'ordenació del desplegable per defecte no es fa diferenciant entre majúscules i minúscules.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal que al modificar un classification schema s'introdueixi a la posició on li toca alfabèticament. També s'ha modificat el mateix fitxer per tal que quan es carrega a l'inici el desplegable, l'ordenació no es faci diferenciant entre majúscules i minúscules.
M4	A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat, hauria de sortir també el goal, ja que a aquest camp se li ha donat molta importància en la documentació del projecte però poca a l'aplicació.	<ul style="list-style-type: none"> S'ha modificat el codi del fitxer <i>PatternsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal que aparegui a la finestra taronja el goal d'un requirement pattern, si és que en té.

Annex 1: Modificacions al codi font de RPToolManager

M5	A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat, les sources queden ordenades de forma aleatòria. Haurien d'aparèixer ordenades alfabèticament.	<ul style="list-style-type: none"> • Ha calgut modificar la classe <i>Source.java</i> del paquet <i>edu.upc.gessi.rptool.domain</i> per tal que implementi la classe <i>Comparable</i> per tal de facilitar l'ordenació de les sources; a conseqüència, s'ha hagut d'afegir el mètode <i>compareTo(...)</i> per tal d'establir quina és la comparació entre dues sources. • S'ha modificat el fitxer <i>PatternsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal que a la finestra taronja dels patrons les sources apareguin ordenades.
M6	A la finestra de patrons, a la informació que surt a la part inferior taronja de la finestra sobre el patró que hi ha seleccionat, l'ordenació dels requirement forms d'aquest és aleatòria, ja que conforme vas modificant l'informació dels forms canvia l'ordenació, i tampoc es correspon amb l'ordenació que hi ha al formulari d'edició del patró.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>RequirementForm.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> per tal que implementi la classe <i>Comparable</i> i s'ha afegit el mètode <i>compareTo(...)</i> que retorna la comparació de dos requirement form. • S'ha modificat el fitxer <i>PatternsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal que a la finestra taronja dels patrons els forms i les extensions d'aquests apareguin ordenats alfabèticament.
M7	A la finestra d'edició de patrons, a l'arbre de navegació del patró, no es veuen ni els forms ni les seves extensions ordenats alfabèticament. Per facilitar la navegació per l'arbre, hauria de ser així.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> el mètode <i>updatePatternTree(...)</i> per tal que l'arbre es carregui i s'actualitzi amb els forms i les seves extensions ordenats alfabèticament.
M8	A la finestra d'edició de patrons, a la pestanya attributes, les keywords no queden ordenades de cap manera i de fet cada cop que es canvia de pestanya l'ordenació canvia, i això és un embolic. S'haurien d'ordenar alfabèticament.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que la llista de keywords aparegui ordenada alfabèticament a la finestra. • També s'ha modificat el mateix fitxer per a que a l'afegir una keyword al patró la llista continuï estant ordenada.

Annex 1: Modificacions al codi font de RPToolManager

M9	A la finestra d'edició de patrons, a la pestanya forms, els requirement forms no queden ordenats de cap manera. S'haurien d'ordenar alfabèticament.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que a la pestanya dels forms aquests apareguin ordenats alfabèticament.
M10	A la finestra per seleccionar keywords (a la qual s'accedeix des de la finestra d'edició de patrons quan es vol afegir una nova keyword), les keywords no queden ordenades de cap manera i això dificulta la cerca d'una en concret. S'haurien d'ordenar alfabèticament.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogSourceList.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogObjects</i> per tal que la llista de keywords aparegui ordenada alfabèticament a la finestra. També s'ha modificat el mateix fitxer per a que a l'afegir una nova keyword o modificar una la llista continuï estant ordenada.
M11	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts, s'hauria d'explicar breument en una banda com funciona el botó <i>Extract from text</i> (dient que els paràmetres s'han de posar entre %...%).	<ul style="list-style-type: none"> Afegir al fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> el codi necessari per tal d'afegir un botó d'ajuda a sota del botó "Extract from text" de les pestanyes Fixed Part i Extensions per tal que surti una nova finestra explicant quin és el comportament del botó "Extract from text". Al mateix fitxer també s'ha modificat el fet que el botó d'ajuda afegit a la pestanya Extension només estigui actiu quan hi ha una extension seleccionada.
M12	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, al desplegable per seleccionar l'extension que volem editar aquestes surten ordenades sense cap mena de criteri, i això dificulta la cerca quan hi ha unes quantes. Haurien d'aparèixer ordenades alfabèticament.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>ExtendedPattern.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> per tal que implementi la classe <i>Comparable</i> i s'ha afegit el mètode <i>compareTo(...)</i> que retorna la comparació de dos extended part. S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que la funció <i>updateExtendedPatternList()</i> ordeni les extensions en funció del seu nom.

Annex 1: Modificacions al codi font de RPToolManager

M13	A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya extended parts, quan es crea una nova extension no queda seleccionada al desplegable.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal que un cop s'ha afegit la nova extension, aquesta quedi seleccionada al desplegable.
M14	<p>A la finestra d'edició de patrons, quan està seleccionat un requirement form i estàs en la pestanya fixed o extended parts, hauria d'haver un acord en el format en el que poden estar els paràmetres d'una fixed o extended part d'un requirement form, ja que:</p> <ol style="list-style-type: none"> Si fem <i>New parameter...</i> deixa introduir el que vulguem com a nom de paràmetre. Si ho fem pel mètode <i>Extract from text</i> no fa ni cas dels paràmetres que tinguin algun caràcter que no sigui una lletra. <p>S'hauria d'unificar l'edició dels noms dels paràmetres en ambdues opcions.</p>	<ul style="list-style-type: none"> S'ha modificat al fitxer <i>FormTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> el codi que cercava els noms de paràmetres al form text per tal que reconegui com a nom qualsevol string que es trobi entre %...%, inclogui el símbol que inclogui ('.', ',', '!', '?', etc).
M15	A la finestra de mètriques, a la informació que surt a la part inferior taronja de la finestra sobre la mètrica que hi ha seleccionada, les sources queden ordenades de forma aleatòria. Haurien d'aparèixer ordenades alfabèticament.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal que a la part inferior taronja de la finestra de mètriques les sources apareguin ordenades.

Annex 1: Modificacions al codi font de RPToolManager

M16	A la finestra de mètriques, igual que a la finestra de patrons hi ha dos botons que es diuen <i>Expand all</i> i <i>Collapse all</i> , en aquesta finestra també hi podrien ser.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>MetricssInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal d'incorporar els botons "Expand All" i "CollapseAll".
M17	Al formulari d'edició de mètriques de tipus float, seria útil indicar quin símbol s'utilitza pels decimals.	<ul style="list-style-type: none"> S'ha afegit al fitxer <i>DialogFloatMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el codi que mostra a la finestra la informació de quin símbol s'utilitza pels decimals.
M18	Al formulari d'edició de mètriques de tipus time point, quan estàs creant una nova mètrica, hauria de sortir la data/hora actual.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que per defecte s'estableixi els valors dels camps <i>Month</i>, <i>Day</i>, <i>Hour</i>, <i>Minute</i> i <i>Second</i> amb el valor actual.
M19	Al formulari d'edició de mètriques de tipus time point, el mes es selecciona per nom. En canvi al camp taronja de la finestra de mètriques, quan selecciones una mètrica de tipus time point, el mes surt com a número. S'hauria d'unificar.	<ul style="list-style-type: none"> S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que l'idioma del selector de mesos sigui anglès i agafi per defecte el mes de gener. S'ha modificat el fitxer <i>MetricsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal que el mes al rectangle taronja de la part de sota de la finestra de mètriques surti amb el nom en anglès.
M20	Al formulari d'edició de mètriques de tipus time point, el mes es selecciona per nom en castellà. En canvi, tota la resta de l'aplicació és en anglès. S'hauria d'unificar.	

Annex 1: Modificacions al codi font de RPToolManager

M21	Al formulari d'edició de mètriques de tipus time point, no surt l'any. De fet, una mètrica de tipus time point no té l'atribut any. S'hauria d'afegir.	<ul style="list-style-type: none"> • S'ha modificat l'estructura de la Base de Dades mitjançant l'eina <i>Squirrel SQL Client</i> per tal d'afegir la nova columna <i>YEAR_1</i> a la taula on es guarden les Time Point Metric. • Degut a la columna afegida a la Base de Dades, s'ha hagut de modificar el fitxer <i>Mapeig.hbm.xml</i> del paquet <i>edu.upc.gessi.rptool</i> per tal que incorporés el mapeig de la nova columna <i>YEAR_1</i>. • S'ha modificat el fitxer <i>TimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.domain.metrics</i> per tal d'afegir el nou atribut <i>year</i> així com els seus mètodes <i>get</i> i <i>set</i>. • S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal de poder mostrar i modificar el nou atribut <i>year</i> a la finestra que crea/modifica una mètrica de tipus time point. • S'ha modificat el fitxer <i>MetricsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal de mostrar l'atribut <i>year</i> d'aquest tipus de mètrica, si és que el té.
M22	Al formulari d'edició de mètriques de tipus time point, seria útil utilitzar un servei de calendari (per les dates) i un desplegable o rellotge per les hores.	<ul style="list-style-type: none"> • S'ha creat un nou fitxer <i>JDateTimeChooser.java</i> al paquet <i>edu.upc.gessi.rptool.gui.common</i> que implementa un calendari i rellotge tipus Windows com el que es vol afegir a la finestra de mètriques de tipus time point. • S'ha modificat el fitxer <i>DialogTimePointMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> per tal que no s'utilitzin els antics selectors per cada camp i s'utilitzi a canvi un nou objecte de la classe <i>JDateTimeChooser</i>.
M23	Al formulari d'edició de mètriques de tipus set, als camps <i>Selected metric</i> i <i>Current metric</i> , es pot seleccionar el valor que tenen, però de fet no és útil aquesta selecció i potser pot portar a confusió. Seria millor si fossin camps bloquejats.	<ul style="list-style-type: none"> • Fixar al fitxer <i>DialogSetMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> l'opció de que no es pugui seleccionar ni a l'arbre de <i>Selected metric</i> ni al de <i>Current metric</i>.

Annex 1: Modificacions al codi font de RPToolManager

M24	Al formulari d'edició de mètriques de tipus set, quan es prem el botó <i>Assign</i> no fa res. S'hauria d'avisar de que s'ha de seleccionar una mètrica simple a la finestra de mètriques.	<ul style="list-style-type: none">S'ha modificat del fitxer <i>DialogSetMetric.java</i> del paquet <i>edu.upc.gessi.rptool.gui.dialogMetrics</i> el mètode de l'acció del botó "Assign" per tal de que si no hi ha cap mètrica simple seleccionada i es prem aquest botó, surti el missatge de que primer s'ha de seleccionar una mètrica simple.
-----	--	---

Noves funcionalitats

Funcionalitat		Solució
F1	Seria interessant poder veure tota la informació del patró en una mateixa finestra (similar a com es mostra a la Taula 1 d'aquest document, a l'apartat 2.3 Estructura d'un patró de requisit).	<ul style="list-style-type: none"> • Ha calgut modificar la classe <i>Keyword.java</i> del paquet <i>edu.upc.gessi.rptool.domain.requirementpatterns</i> per tal que implementi la classe <i>Comparable</i> per tal de facilitar l'ordenació de les keywords; a conseqüència s'ha hagut d'afegir el mètode <i>compareTo(...)</i> per tal d'establir quina és la comparació entre dues keywords. • S'ha modificat el fitxer <i>PatternsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal de crear els mètodes necessaris que retornen el codi html corresponent a la taula amb tota la informació d'un patró. • S'ha modificat el fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal que al fer doble clic sobre un node que es correspon amb un patró, mostri una nova finestra amb tota la seva informació.
F2	Seria interessant poder veure tota la informació d'una mètrica en una mateixa finestra.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>MetricssInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal que al fer doble clic sobre un node que es correspon amb una mètrica, mostri una nova finestra amb tota la seva informació. • S'ha modificat el fitxer <i>MetricsNode.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFrameMetrics</i> per tal de crear els mètodes necessaris que retornen el codi html corresponent a la taula amb tota la informació d'una mètrica.
F3	Seria útil poder fer una còpia d'un classification schema sencer.	<ul style="list-style-type: none"> • S'ha modificat el fitxer <i>MediatorPatterns.java</i> del paquet <i>edu.upc.gessi.rptool.storage</i> per tal d'afegir dues noves operacions: una que permeti recuperar els patrons que pertanyen a un basic classifier determinat i una altre que només faci update d'un patró. • S'ha modificat el fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal d'incorporar un nou botó que permeti fer una còpia del classification schema seleccionat.

F4	<p>En la finestra d'edició de patrons, seria útil tenir una nova pestanya quan el patró està seleccionat en l'arbre de navegació de la finestra, per poder gestionar a quins classificadors pertany un patró dins d'un classification schema.</p>	<ul style="list-style-type: none"> • S'ha creat la nova classe <i>MarkedClassifierNode.java</i> al paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> que implementa els nodes marcats de l'arbre que dibuixa l'esquema seleccionat en la pestanya Classification Schemas quan es crea/modifica un patró. • S'ha creat la nova classe <i>TreeSelectedSchema.java</i> al paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> que implementa l'arbre que dibuixa el classification schema seleccionat en la pestanya Classification Schemas quan es crea/modifica un patró. • S'ha creat la nova classe <i>SelectedSchemaTreeCellRenderer.java</i> al paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> que implementa com dibuixar l'arbre de l'esquema seleccionat en la pestanya Classification Schemas quan es crea/modifica un patró. • S'ha modificat el fitxer <i>MediatorSchemas.java</i> del paquet <i>edu.upc.gessi.rptool.storage</i> per tal d'afegir una nova operació que permeti recuperar de la Base de Dades un internal classifier donat el seu identificador. • S'ha modificat el fitxer <i>MediatorPatterns.java</i> del paquet <i>edu.upc.gessi.rptool.storage</i> per tal d'afegir una nova operació que permeti recuperar de la Base de Dades els identificadors dels interns classificadors als que pertany un patró concret. • S'ha modificat el fitxer <i>PatternsInternalFrame.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns</i> per tal d'enviar a la finestra <i>DialogPattern</i> el path que hi ha seleccionat en l'arbre en el moment en que es crea la finestra. • S'ha modificat el fitxer <i>DialogPattern.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de: <ul style="list-style-type: none"> ○ Guardar el path de l'arbre de la finestra de patrons des d'on es crea la finestra i enviar aquest path a l'entitat <i>PatternTabs</i> quan es crea. ○ S'ha creat un nou mètode <i>getClassifierParent()</i> que retorna l'Internal Classifier que hi havia seleccionat quan s'ha creat la finestra <i>DialogPattern</i>. ○ S'ha modificat el mètode <i>okAction()</i> per tal que actualitzi els internal classificadors als que pertany el patró amb els que s'han seleccionat en la pestanya Classification Schemas. • S'ha modificat el fitxer <i>PatternTabs.java</i> del paquet <i>edu.upc.gessi.rptool.gui.internalFramePatterns.dialogPatterns</i> per tal de:
----	---	--

Annex 1: Modificacions al codi font de RPToolManager

		<ul style="list-style-type: none">○ Guardar el path de l'arbre de la finestra de patrons des d'on es crea la finestra <i>DialogPattern</i>.○ Crear la nova pestanya Classification Schemas.
--	--	--